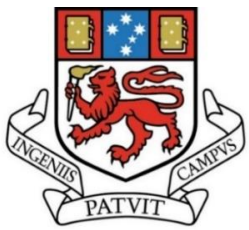


UNIVERSITY
OF TASMANIA

A Hybrid Failure Diagnosis and Prediction Framework for Large Industrial Plants

By

Dohyeong Kim

A dissertation submitted to the School of Technology, Environments and Design, University of Tasmania in fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Tasmania

July, 2018

Declaration

This thesis contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information and duly acknowledged in the thesis, and to the best of my knowledge and belief no material previously published or written by another person except where due acknowledgement is made in the text of the thesis, nor does the thesis contain any material that infringes copyright.

Dohyeong Kim

_____

July, 2018

Authority of Access

This thesis may be made available for loan and limited copying and communication in accordance with the Copyright Act 1968.

Dohyeong Kim

July, 2018

Abstract

A Hybrid Failure Diagnosis and Prediction Framework for Large Industrial Plants

Modern industrial plants contain large number of facilities interacting with thousands of sensors and control. A single failure in a facility can produce inconsistent outcomes, which can be affected to core part of industrial plant, and it becomes a critical industrial disaster. Therefore, it is crucial to find and apply the best solution for maintaining facilities and preventing industrial disasters. The early-stage solution was the regular maintenance but this approach cannot be a perfect solution to prevent most industrial disasters. This is because regular maintenance is not effective for all but only few facilities, and is spent too much time and cost to afford. The recent trend of industrial plant maintenance focuses on two main factors, alarms and human expertise.

The system collects the status of different types of facilities from the sensors, which are attached, on each facility. If there is any specific symptom detected from sensors, the alarm will be ringed. The collected alarm data are sent to the human experts in the real time. The human experts have experienced various types of industrial disasters so they have sufficient knowledge in diagnosing and treating failures.

In this dissertation, I studied how to use alarm data and expert knowledge together with these characteristics. In this study, I constructed knowledge using failure reports reflecting alarm data, expert knowledge, which are significant knowledge resources of the industrial field and proposed a method to continuously manage and use such knowledge.

This dissertation can be divided mainly into three parts of subjects of researches.

In the first study, I propose a hybrid knowledge engineering method based on machine learning- expert knowledge, which enables machine learning and domain experts to generate and update knowledge together.

First of all, after constructing a knowledge base by applying real-time alarm data and machine learning, the expert can directly update the knowledge continuously, thereby enabling knowledge creation and management in a fast and efficient manner.

After constructing a knowledge base by applying real-time alarm data and machine learning, the expert can directly update the knowledge continuously, thereby enabling knowledge creation and management in a fast and efficient manner.

Second, I propose a methodology for constructing causal knowledge as overall conditions and treatment actions for failure. Failure report includes the cause-and-effect relationship and its order of occurrence. The proposed methodology analyzes the failure reports written by domain experts using natural language processing techniques and helps to organize the cause-effect and treatments for the failures into a network form.

Finally, the knowledge constructed by the hybrid knowledge engineering method and the causal failure knowledge are fused and applied to the fault diagnosis and prediction.

As a result of the performance analysis, the proposed framework is superior to the other methodologies regarding failure diagnosis and prediction. The proposed decision support method in this dissertation can evolve the two types of knowledge required in the field gradually. Thus it was able to solve the knowledge management difficulties, and using the two knowledge together complementarily; knowledge management efficiency has been achieved. Moreover, it showed superior performance compared to existing methods based on data.

Key words:

Sensor Data Mining, Machine learning, Knowledge-based System, Expert system, Knowledge Engineering, Knowledge Reuse, Failure Detection, Failure Prediction, Preventive maintenance.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

Modern industrial plants contain large number of facilities interacting with thousands of sensors and control. A single failure in a facility can produce inconsistent outcomes, which can be affected to core part of industrial plant, and it becomes a critical industrial disaster. Therefore, it is crucial to find and apply the best solution for maintaining facilities and preventing industrial disasters [1]. The early-stage solution was the regular maintenance but this approach cannot be a perfect solution to prevent most industrial disasters [2]. This is because regular maintenance is not effective for all but only few facilities, and is spent too much time and cost to afford. The recent trend of industrial plant maintenance focuses on two main factors, alarms and human expertise. The system collects the status of different types of facilities from the sensors, which are attached, on each facility. If there is any specific symptom detected from sensors, the alarm will be ringed. The collected alarm data is sent to the human experts in the real time. The human experts have experienced various types of industrial disasters so they have sufficient knowledge in diagnosing and treating failures. Applying facility sensor network, alarm data and human expertise seems to be a good combination in handling failure but this approach also has two major issues.

Firstly, the system may produce alarm flooding. The amount of the collected alarm is too enormous to be properly checked and handled by human experts. Owing to this, some severe failures can be misled or skipped, which may cause a critical industrial disaster. Secondly, diagnosis and treatment activities are too depended on human experts. There are only limited numbers of human experts who have sufficient experiences in the certain industrial plant. It cannot be expected any proper treatment if human experts are not available. Additionally, some failure cannot be diagnosed or treated since the expert have never experienced before [3].

In order to solve those issues, knowledge based systems were mainly constructed by using

two different approaches, machine learning technique and human expertise. For the first solution, machine learning has been applied in order to manage knowledge for detecting failures. Machine learning techniques enable the system to acquire the knowledge from existing alarm data with no help of a domain expert. The techniques are very fast in finding important pattern and knowledge from the provided data so it reduced the time and cost. However, machine learning has some drawbacks, such as over-generalization and overfitting [4].

Another solution for failure detection knowledge based system was conducted with human experts. Human domain experts have enough experience so they can save knowledge in order to solve complex problems in a specific domain. However, knowledge acquisition from a human expert is normally in a slow pace. Even if the knowledge was acquired, the acquired expertise tends to be lopsided and would not cover the whole concept of knowledge in the domain since experts acquire domain knowledge based on their past experience [5].

In this research project, I focus on discovering the failure detection knowledge, and preventing the failure in the large industrial plants.

Throughout this project, I focus on the following three studies.

The first study is to discover the failure detection knowledge by using real-time alarm data and machine learning techniques. It analyses the real time nature of alarm data, and its characteristic. Based on the analysis result, the best features can be selected to discover the knowledge with different machine learning techniques. This study will provide a framework to identify the machine-learning based knowledge for the failure detection in the large dataset.

The second study is to acquire failure detection knowledge from continuous alarm data, and maintaining human expertise. The study focuses on proposing a new failure detection approach with machine learning and human expertise in a large and continuous domain.

The third study is to acquire the failure prediction knowledge from domain expert written

failure reports. This study aims to extract the domain expertise by using natural language processing technique, and store this with the network-based knowledge base.

1.2 Motivation

Dealing with alarm data and analysing expertise is the main key to manage the large industrial plants. A CEO of Tesla, Elon Mask, has made a statement about the necessary of analysing alarm data and understanding industrial expertise after the huge Tesla industrial accident:

"In order to prevent the huge industrial accident, it is crucial to acquire real-time facility data and analyse the expertise, and computerise them for the intelligent system"

This statement well represents the motivation of this research goal, proposing failure status detection and prediction system by using real-time alarm data and analysing expertise from the failure status report. The following three research questions presents the main point I would like to solve in order to achieve the main research goal.

Research questions:

A. How to discover the failure detection knowledge from machine learning technique

a. How to find/select the best feature from alarm data in the large industrial plant

b. How to detect the pattern of failure

B. How to maintain the machine learning based failure detection knowledge base with human expertise

a. How to build the maintainable machine-learning-based knowledge base from the continuous alarm data.

b. How to update the existing knowledge base with human expertise.

C. How to extract the computer usable knowledge from human-written failure reports written in technical term and restricted representation

a. How to extract failure knowledge from human-written failure reports

b. How to model/represent the extracted knowledge

c. How to identify the cause and effect of failure

1.3 Contribution

The main work of this study can be divided into three parts. First, characteristics of real domain alarm data were analyzed through analysis. Second, I proposed a knowledge acquisition method to overcome the domain characteristic of continuously changing alarm knowledge.

Finally, but it accumulated in the field, extracting the failure reports which unable to use in the system, and proposes methods to sustainable management. The contribution for each part is briefly shown below.

A. Hybrid Knowledge acquisition method

In the first study, I propose a hybrid knowledge engineering method based on machine learning- expert knowledge, which enables machine learning and domain experts to generate and update knowledge together. This is a new approach that combines existing machine learning with expert knowledge-based methods. First of all, after constructing a knowledge base by applying real-time alarm data and machine learning, the expert can directly update the knowledge continuously, thereby enabling knowledge creation and management in a fast and efficient manner. Also, it shows that knowledge base construction using machine learning through experiments is equal or higher than that of existing methods and also it is proved that experts can update the knowledge base and improve performance accuracy.

B. Causal knowledge representation

Second, I proposed a methodology for constructing causal knowledge as overall conditions and treatment actions for failure. The proposed methodology analyzes the failure reports written by domain experts using natural language processing techniques and helps to organize the cause-effect and treatments for the failures into a network form. Unlike existing ontology engineering method, this method extracts the knowledge through analyzing the failure reports based on NLP methods automatically. In addition, an access is achieved to similar knowledge by constructing the network based on the similarity between knowledge. Also, usability was also improved by developing GUI-based knowledge engineering tools.

C. Failure prediction using hybrid knowledge representation

Finally, the knowledge constructed by the hybrid knowledge engineering method and the causal knowledge are fused and applied to the fault diagnosis and prediction. Unlike the existing data-driven method, expert knowledge is reflected, and it can help the decision making of experts by presenting the process of deriving the answers and examples. Also, by using single knowledge representation method for representing relationships between complex phenomena in order to failure diagnosis, it is possible to update expert knowledge unlike existing methods with high knowledge management cost, and also the knowledge management is very efficient. In addition, the performance shows that the proposed framework is superior to other methodologies regarding failure diagnosis and prediction.

1.4 Thesis Organization

The proposed methodology can be found in Figure 1.1, which includes each process independently. As can be seen in the Figure, there are two types of data, including alarm data and failure report, that we collected. The alarm data was collected from the Hyundai Steel corporation, and built the knowledge base with hybrid knowledge engineering approach for failure detection. The failure report written by experts in the factory were applied to build the casual knowledge process map, which describes the cause-and-effect relationship. This process map is linked to the developed knowledge base in order to predict the future failure.

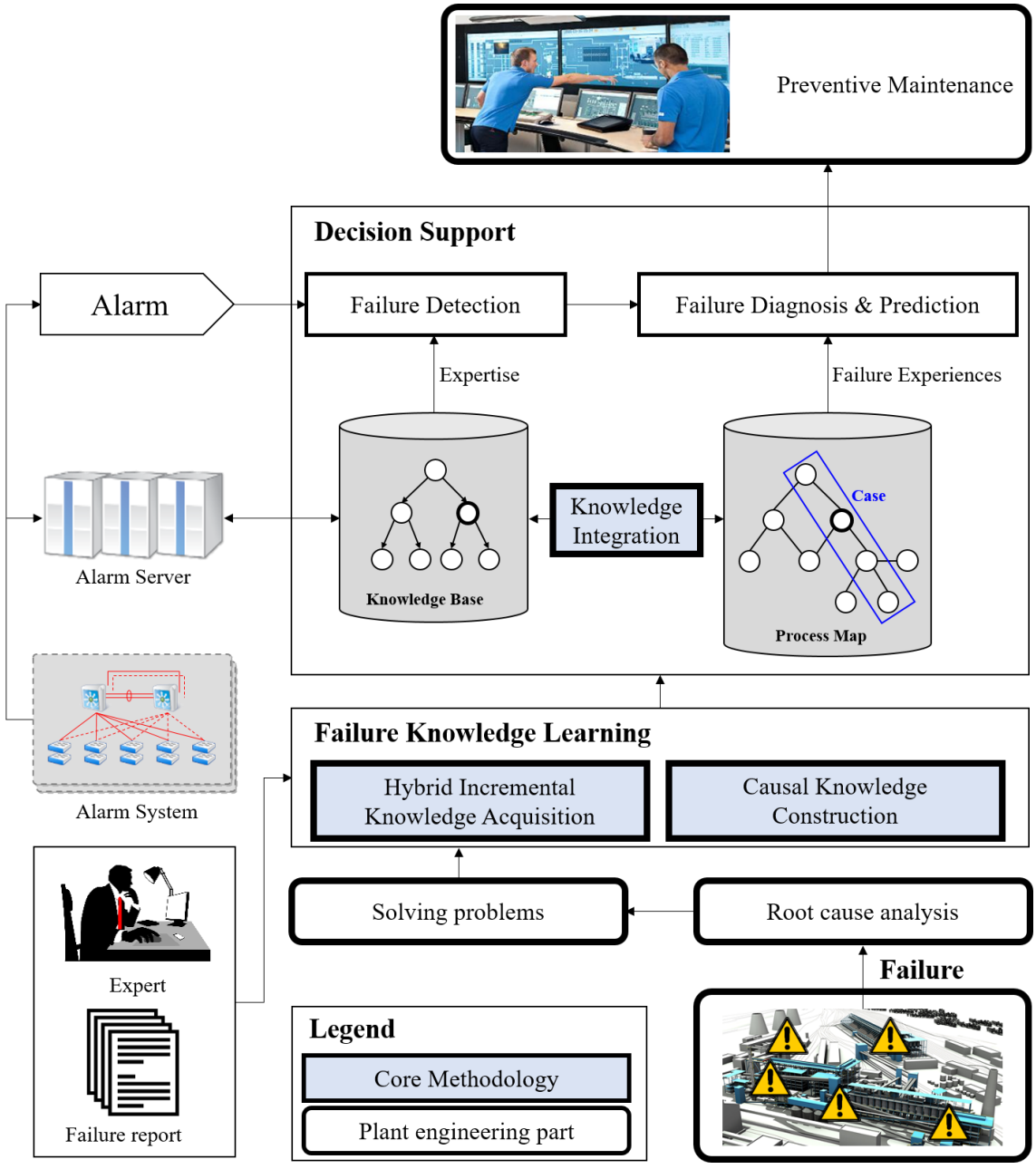


Figure 1.1 Overview of the Proposed Failure Detection and Prediction Framework

Figure 1.2 shows the dissertation organisation, and the following list contains the brief explanation of the contents of this dissertation.

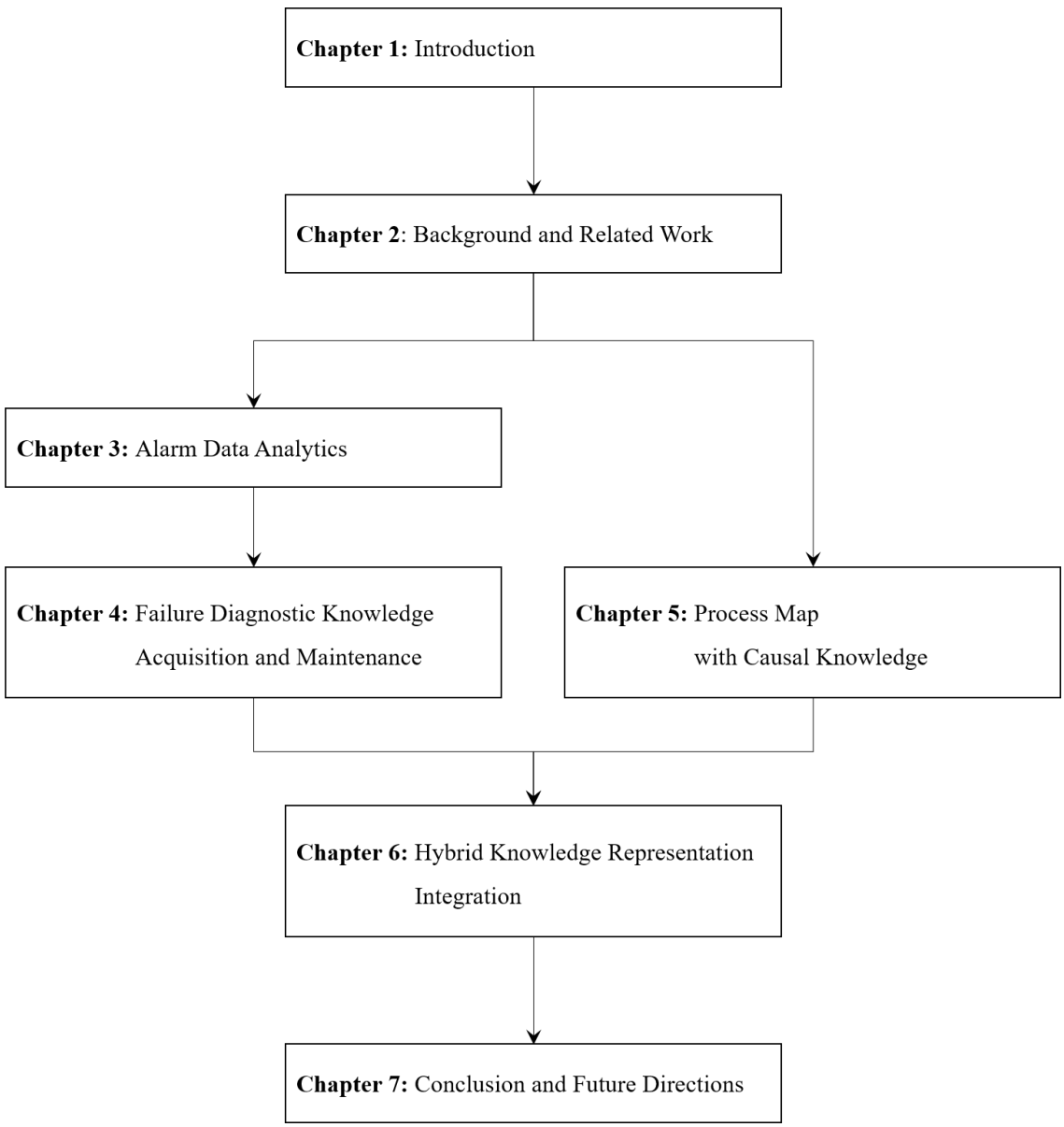


Figure 1.2 Thesis Organization

Chapter 1: Introduction

In this chapter, I describe the research background, research goals, research question and contribution, and introduce the proposed framework.

Chapter 2: Background and Related Work

This chapter introduces background knowledge and related research trends of this study.

Chapter 3: Alarm Data Analytics

This chapter shows the analysis results of the alarm data collected from the real domain used in this study. The developed alarm collection interface and alarm occurrence patterns for a specific period can be checked, and it also provides feature selection and labeling method and results from data.

Chapter 4: Failure Diagnosis Knowledge Acquisition and Maintenance

In this chapter, I propose a method of constructing failure knowledge using alarms for failure diagnosis. Unlike the existing method, I have proposed a novel hybrid method that can construct an initial knowledge base by applying machine learning to data, and experts can continuously modify the knowledge on changing knowledge. For the performance measurement, the proposed methodology compares the accuracy of the typical machine learning methods.

Chapter 5: Process Map with Causal Knowledge

In this chapter, I propose a method to construct a network type of knowledge base by extracting structural causal knowledge from the failure report that describes the failure occurrence

and solution process for the failure situation. The proposed method uses natural language processing method to convert the failure report into structural knowledge automatically and automatically builds up the relationship between the precise analyzed knowledge and network-based knowledge. It also introduces applications developed to enable users to add and optimize their knowledge directly using similar knowledge recommended by the system.

Chapter 6: Hybrid Knowledge Representation Integration

This chapter introduces the system for preventive maintenance at the factory, using the knowledge to detect facilities' failures built from alarms and domain experts in chapter 4 and the failure cases built in chapter 5. It shows the performance evaluation results by comparing the methods to integrate the two knowledge and the existing fault prediction method.

Chapter 7: Conclusion and Future Directions

This chapter shows the conclusions and significant contributions to the study and concludes the thesis.

2 Background and Related Work

2.1 Knowledge Representation Techniques

This chapter explains how to represent knowledge, and is divided into five sections according to knowledge representation type. Each will be described in turn.

A. Declarative Knowledge Scheme

- Logic is the traditional and most representative method.
- It is the most common way to deduce new facts and is suitable for small and simple problem areas. It is not appropriate to express real-world knowledge because the reasoning becomes complicated.

B. Procedural Representation Scheme

- Typically, it is a rule-based system, and knowledge is expressed as a rule. Many expert systems are implemented on a rule basis.
- As a form that is easy for a person to understand or express, a Rule takes the form of "If-Then" and consists of a conditional statement and a conclusion part.

C. Network Based Knowledge Representation Scheme

- Semantic net is a representative network type of knowledge representation method.
- It is a directional graph expressed by edges representing relationships between concepts, and nodes representing concepts. It is a system embodied in the form of modeling a method of reminding human memory.

D. Structured Knowledge Representation Scheme

- It is a structural extension of the network knowledge representation method. Typically, there are Frame, Object-oriented, and Script. Frame is the most common.

E. Ontology

- It is a method of constructing a knowledge base by giving meaning to information resource and expressing relationship between information in a graph.

2.1.1 Logic

An intelligent system based on artificial intelligence or a knowledge based system that accumulates knowledge by expressing knowledge in the knowledge base in order to solve a given problem and it basically includes a reasoning function that can be used. In other words, this system expresses knowledge by symbolizing and translating. It is through reasoning that knowledge can be utilized.

Logic is a traditional form of knowledge representation, providing a format that allows all statements to be true and false, and enables automation of reasoning. It expresses the concept of logic naturally based on mathematical grounds, and is useful for formalizing knowledge through mathematical proof. That is, it is easy to add and delete knowledge that is expressed simply. However, it is difficult to express procedural knowledge, and is difficult to express practical complex knowledge because of a lack of the constitutive law of facts.

Logic-based knowledge representation is divided into propositional logic and description logic. Propositional logic is logic that is based on sentences that can judge true or false, and is used to express facts about the real world. Predicate logic is an extension of the propositional logic. It can represent the grammatical structure and meaning of a sentence, not merely discussing true and false. Variables and quantifiers can be used to represent an unspecified number of concepts and express more complex facts than propositional logic.

Table 2.1 shows symbols for representing logic, and Table 2.2 is an example for proposition logic and description logic.

Table 2.1 List of Logical Symbols

Type	Symbol	Definition
Punctuation	“(“, “)”, “, “	
Connectives	¬	Logical negation
	∧	Logical conjunction (“and”)
	∨	Logical disjunction (“or”)
	∃	“there exist …”
	∀	“for all…”
	=	Logical equality
Variables	*Eg. x, y, z*	An infinite supply of symbols

Table 2.2 Examples of logic knowledge representation

Division	Examples	Expression
Propositional logic	P exists P is q	p P → q
Description Logic	X exists establishing p(x,y) for all y	∃x（∀y p(x,y))

2.1.2 Rule

Rules are the easiest form to express knowledge. It has an "IF-THEN" structure and can be expressed as an IF (conditional statement) THEN (conclusion part).

Example: IF is (A and B), THEN it is C.
That is, A and B is C.

It is suitable for the knowledge required to make a decision or conclusion, and is easy to express intuitive knowledge of human. Because of this, it is utilized in many knowledge-based systems. Knowledge is represented by a single rule, so it can be represented as a module. While addition and deletion are easy, it is expensive to analyze and define the problem, and the complexity may increase.

Figure 2.1 is an example of a general rule-based system. The main functions of the system are as follows: It is the most important function that constitutes a rule-based expert system, and various components can be added as needed.

- Knowledge base: The knowledge expressed in IF-THEN form is expressed and saved. If the condition of the rule is satisfied for a given input, it is said that the rule is fired, and then the conclusion is performed.
- Database: The input data is contained from the knowledge base to obtain the answer.
- Inference engine: The inference engine plays a role in reasoning in the expert system for the given input, and the inference is conducted using the knowledge base.
- Explanation facilities: It is the function of explaining the process of knowledge to the user; that is, the reasoning process or the validity of the derived answer.
- User interface: It is a communication means for the user to utilize the expert system and pursues the form as easy and convenient as possible.
- Developer interface: debugging tools, knowledge editing, and I/O interface interworking are included for system developers.

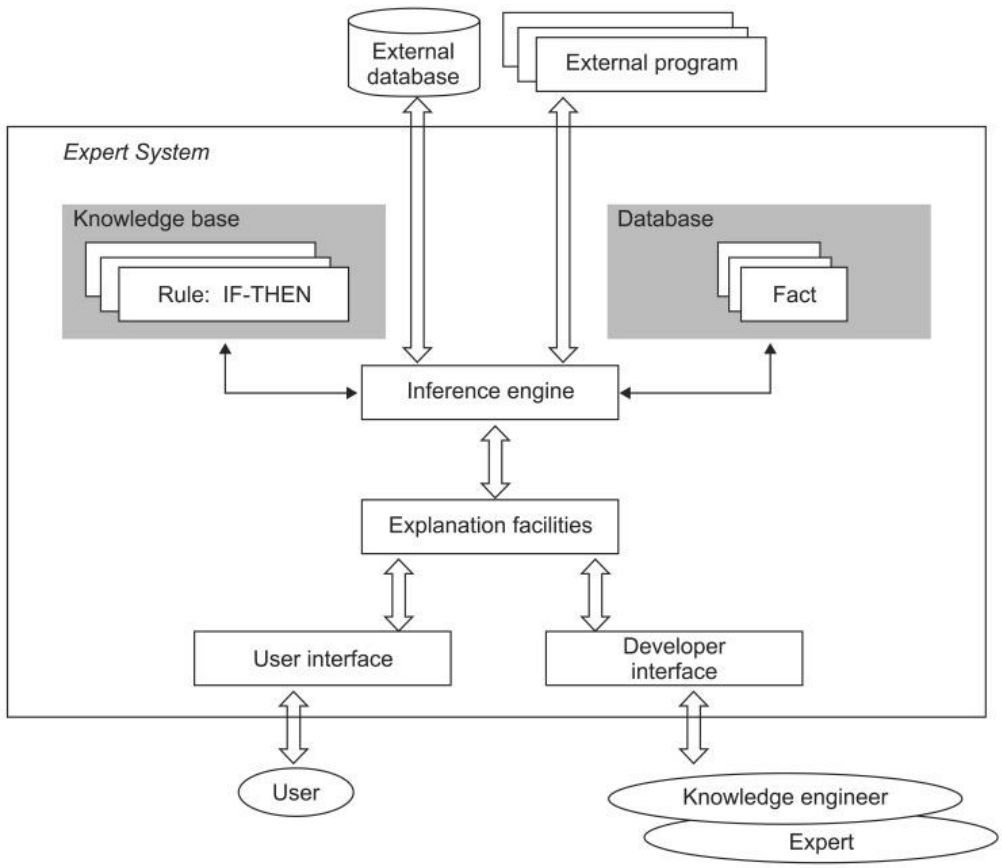


Figure 2.1 The conceptual diagram of traditional rule-based systems

2.1.3 Semantic Network

Semantic net is a directional graph expressed as nodes and edges. A node corresponds to a concept, an idea, an action, a state, or an assertion, and an arc expresses a relationship between nodes. In 1968, M. Ross Quillian developed a graph for knowledge representation and reasoning, used as a method of modeling and embodying a way to relate human memory. Figure 2.2 shows an example of a semantic network.

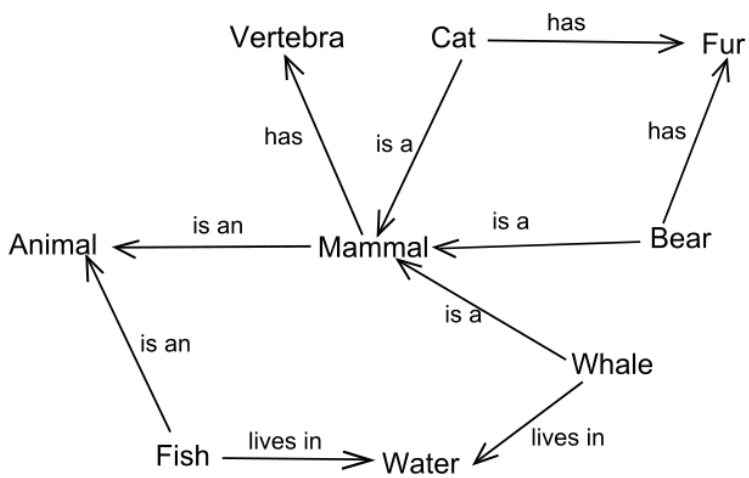


Figure 2.2 An example of semantic network[1]

The semantic network is inherited from an upper node to a lower node. To identify the meaning of these inheritance relationships, the arc can be represented by multiple labels to indicate relationships between nodes.

(Is-a) and (Has label) are the most common expressions in the semantic network model. In the example shown in Figure 2.2, it is easy to deduce that Cat is a Mammal through (Is-a) label. The (Has label) also indicates proprietary relations, and it means the Bear has Fur. Thus, a

[1] https://en.wikipedia.org/wiki/Semantic_network

semantic network is capable of expressing reasoning or complex reasoning. However, there is a disadvantage that the knowledge base is getting larger and cannot be handled when the network structure becomes complicated.

Wordnet is an example that used a typical semantic network. It was firstly developed as an English vocabulary dictionary, and it provides vocabulary classification and brief meaning by constructing a synonym dictionary called ‘synset’. In addition, It can be easily recognized the usage of language by defining the relation between meanings as various relations as shown in Figure 2.3. It is developed for various languages as well as English. It is used in the field of Natural Language Processing (NLP) and is applied for textual interpretation and artificial intelligence purposes.

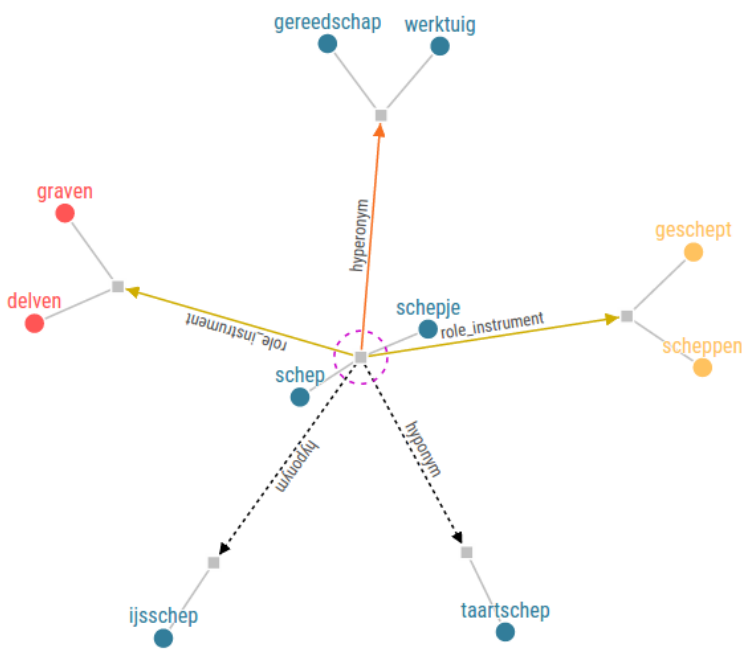


Figure 2.3 An example of Synset visualization[2]

[2] http://cornetto.clarin.inl.nl/help/help_visualization.html

2.1.4 Frame

The frame is a knowledge representation model proposed by Marvin Minsky in 1974 and is a method used for modeling knowledge based on human memory and cognitive processes. It can be said that the knowledge structure is suitable for describing the overall situation or object, the attribute of the object, and expressing the relationship between the objects. Therefore, it is called an object-oriented representation.

The frame structure consists of a frame name and a slot. A slot consists of a slot's name and a slot's value, and a frame can contain several slots. In order to have the relationship between the slots, as shown in the Figure 2.4, another frame can be inherited by specifying another frame in the slot.

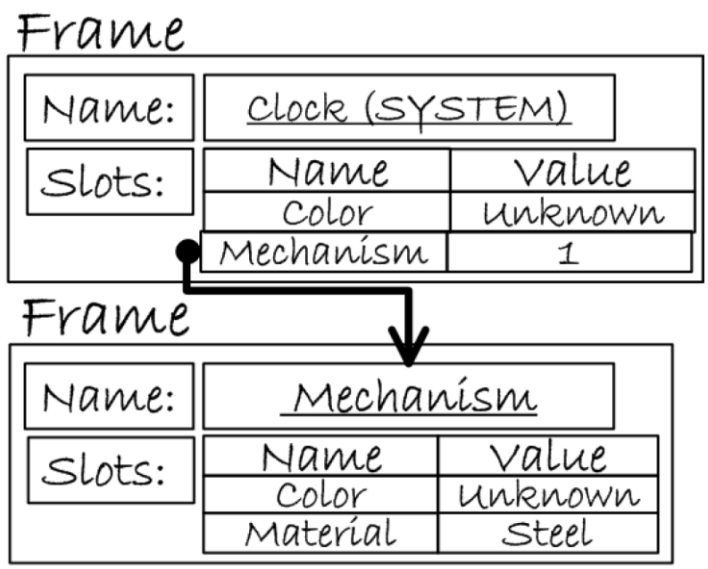


Figure 2.4 An example of frame–based representation [6]

2.1.5 Semantic Web

The Semantic Web was presented by Tim Berners-Lee in 2001 as a vision of Web technology. The Semantic Web expresses the information about resources (web documents, various files, services, and etc.) existing in a distributed environment such as the Internet in a structured form (ontology), and means a framework that allows automated machines such as computers to process them.

Currently, the web is based on HTML, and it can be said that computers are not meta-data that can understand web information, but rather information that is easy for humans to understand. For example, if there is a tag " Snow </ em> White </ em>," the computer will only highlight the two words 'Snow' and 'White.' It cannot be found any relations between two words.

On the other hand, the Semantic Web is built on a markup language based on XML. RDF (Resource Description Framework) is a language that can represent metadata, and it expresses the concept as a triple form <Subject, Predicate, and Object>. The RDF example above would be expressed as < urn: Snow, urn: Color, urn: White>. When a computer interprets a sentence written in triple form, 'Snow' has 'Color' as 'White.'

The Semantic Web expresses meanings on the web as a graphical ontology based on this triple structure. The hierarchical structure of semantic web technology is as follows as shown in Figure 2.5:

- URI (Uniform Resource Identifier): Represents the name, location, etc. of an object to identify a web resource
- IRI (International Resource Identifier with UNICODE):
- XML (extensible Markup Language): Meta information, representation, language and standard.
- RDF: Structural representation of information or information resources
- RDFS: Schema information of RDF, representing a lightweight ontology

- SPARQ: Query language of RDF
- RIF: Hierarchy for defining and exchanging rules
- OWL: Common understanding and concept shared for a specific domain, language for expressing the relationship between concepts
- Logic: Ability to draw new conclusions based on existing information
- Proof / Trust: Trust about the Web

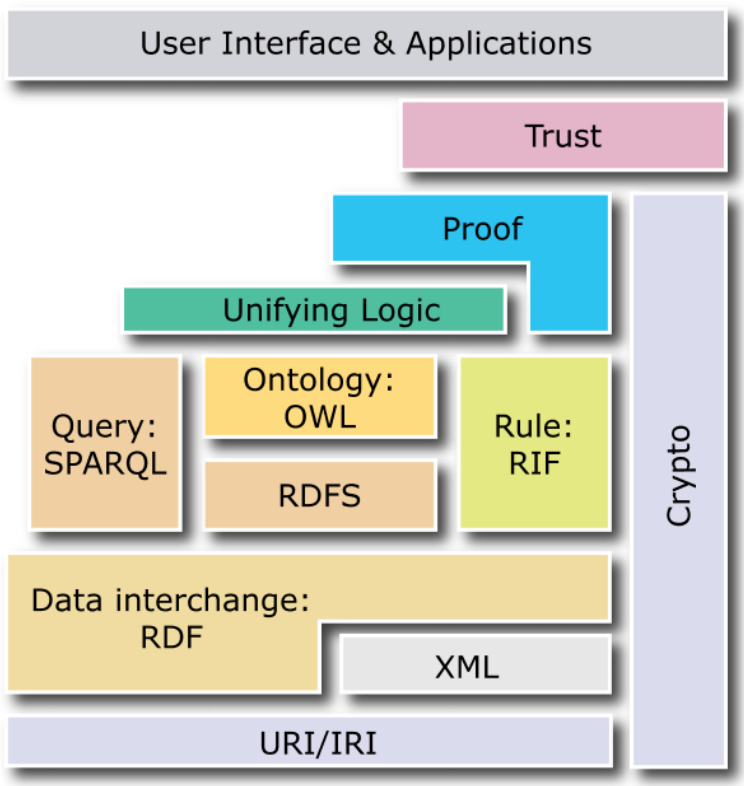


Figure 2.5 Layers of Semantic Web Technology[3]

[3] https://www.w3.org/2007/03/layerCake.png

2.2 Machine Learning Techniques

Section 2.2 includes different type of machine learning techniques that applied to knowledge engineering research, such as detection, prediction, or recommendation. Machine learning is a field of artificial intelligence that studies algorithms and techniques that computers can learn. For example, when there is a large amount of information about data flow in the network, it can be distinguished from abnormal infiltration. Machine learning is the key to representation and generalization. Representation is a data evaluation, and generalization is processing of data that has not yet been confirmed. Learning and classification operations in machine learning are illustrated in the Figure 2.6.

Training data is generalized, refined, and trained with specific machine learning algorithms. In this process, a knowledge model is constructed. When this knowledge model is used, the test data or the new input data is substituted into the knowledge model, and the answer is found through the classifier to find the solution. The main machine learning algorithms, K-Near Neighbor, Decision Tree, Naïve Bayes Theorem, Support Vector Machine and Neural Network, are introduced in each section.

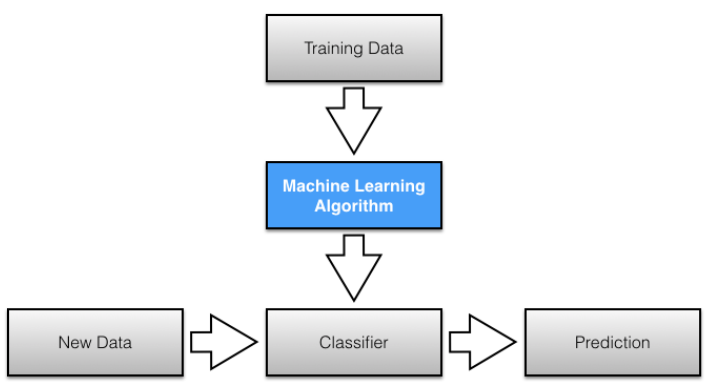


Figure 2.6 Traditional machine learning process[4]

[4] http://sebastianraschka.com/Articles/2014_intro_supervised_learning.html

2.2.1 KNN

KNN (K - Near Neighbor) is a representative unsupervised learning method introduced by Altman in 1992 [7] and is mainly used for classification or regression. That is, the data is classified according to the characteristics without specifying a prediction result in the data. K is the number of neighbors around the reference data. In the example as shown in Figure 2.7, the square and triangle are located around the circle data, so K will be 3.

This method is suitable when the data groups have data with a homogeneous tendency, and therefore it is not effective unless there is a clear difference between the groups. The main feature is that this method is the simplest and fastest effective learning method, but since the distance between all the data must be calculated, the amount of data increases the learning cost.

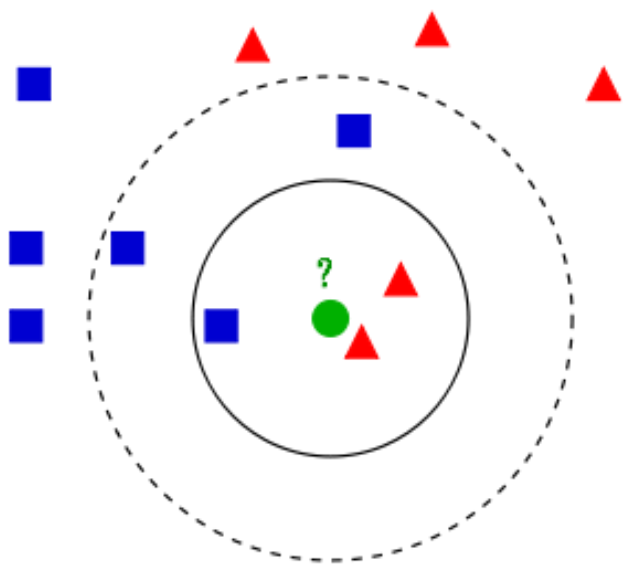


Figure 2.7 An example of K-NN Classification

2.2.2 Decision Tree

Decision Tree as shown in Figure 2.8 is a commonly used learning method in the field of data mining. A knowledge model, which is a tree structure, is created through a label, which is a prediction result already defined from a specific attribute of the input data. In the tree structure, each node corresponds to one attribute, and the branch to the child node corresponds to a possible value of attribute. The advantage of this model is that it is easy to interpret and understand the results, and works well with large datasets. The disadvantage, however, is that if a learner cannot generalize training data properly, it can create too complex of a tree. This is called over-fitting, and a pruning method that eliminates branches that lower knowledge accuracy should be used together in the learning process. The most representative algorithms are ID3, C4.5, C5.0, and CART.

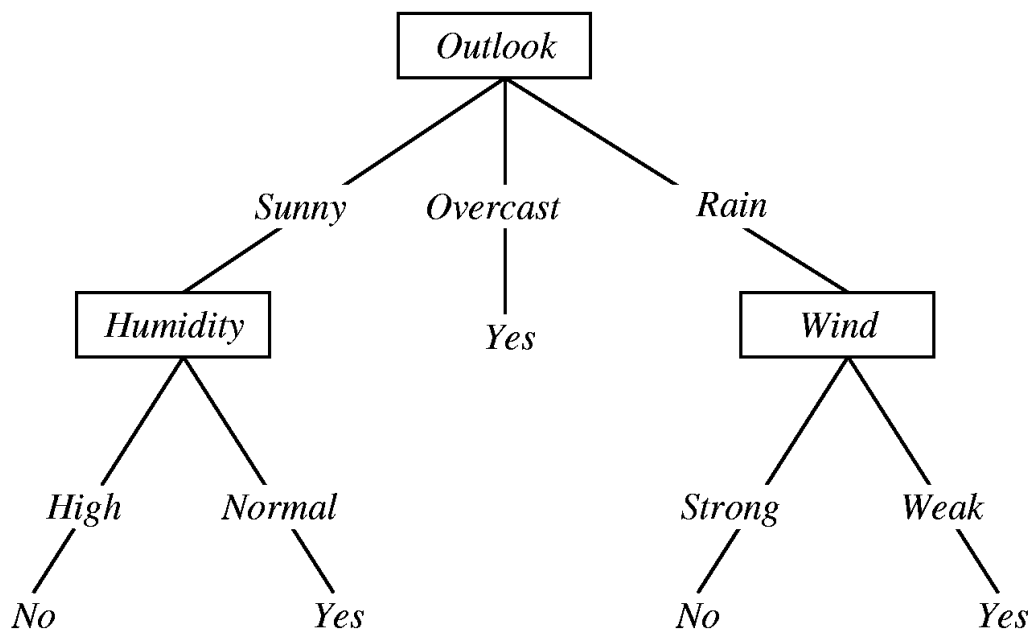


Figure 2.8 An example of Decision Tree Classification

2.2.3 Naïve Bayes Theorem

Naïve Bayes has been extensively studied in the field of machine learning since the 1950s and is a kind of probability classifier applying Bayes' theorem that assumes independence from the properties. Naïve Bayes is a supervised learning method and the simplest way to create a classifier is to train on the basis of a general principle, not a single algorithm. The main principle of this method is that all property values are independent of each other. It is assumed that there are no associations between properties that can classify a particular entity in the classifier, and each contributes independently to classify entities among the properties. The most important feature of this algorithm is that it works well even if the amount of training data is small even though it is relatively simple and simple assumption.

Conditional probability means the probability that event B occurs when event A occurs. In this case, the conditional probability P(B|A) implies the probability that events A and B occur at the same time.

$$\mathrm{P(B|A)} = \frac{P(A \cdot B)}{P(A)}$$

The Naïve Bayes divider classifier in Naive compares the probabilities belonging to a specific class when an element contained in the data is found, and selects the class having the highest probability.

2.2.4 Support Vector Machine

SVM (Support Vector Machine) as shown in Figure 2.9 is used for classification and regression analysis as a supervised learning machine learning method. The SVM creates a probabilistic linear classification model that can classify new data into categories based on data categories given the data belonging to one of the two categories of data. Linear as well as nonlinear classification is possible. The data category classifies the data characteristics by a margin that represents the distance of the data with the hyperplane as a boundary. Typical uses of SVM include recognition of handwriting characteristics, classification of proteins from compounds in medicine, image classification, and so on.

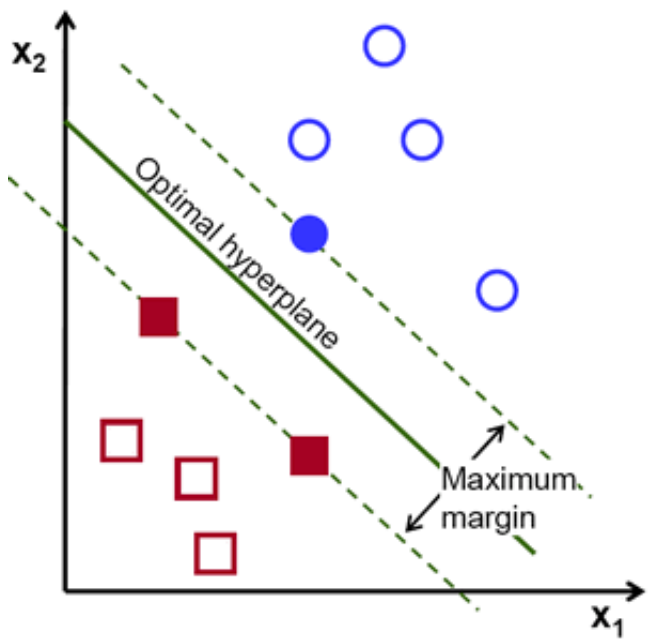


Figure 2.9 An example of SVM[5] Classification

[5] https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

2.2.5 Neural Network

ANN (Artificial Neural Network) is a statistical learning method that imitates biological neural networks in machine learning and cognitive science. As shown in Figure 2.10, ANN refers to a model in which artificial neurons (nodes) forming a network of synapses changes their defective strengths of synapses through learning and has problem-solving ability. ANN can operate in two ways: supervised learning and unsupervised learning. When other machine learning methods operate on a rule basis, they are often used for learning through unstructured data such as image or speech recognition, which is difficult to solve with this method.

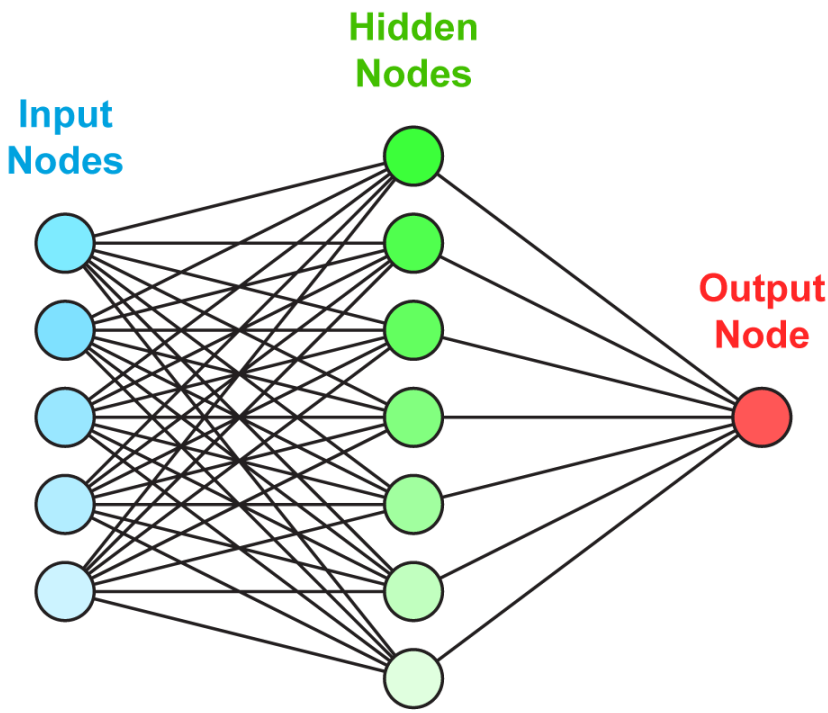


Figure 2.10 A conceptual diagram of Neural Network Classification[6]

[6] http://www.butleranalytics.com/neural-network-data-mining-explained/

2.3 String Comparison

Section 2.3 presents the outcome of the literature review in terms of entity/string comparison. The section focused on reviewing several string comparison techniques that can be used in the process map development.

In this section, I will briefly introduce how to compare similarities and strings between texts. String comparisons have long been studied in the field of computer science and are widely used in information retrieval and text mining. Representative methods are 'Edit-Distance,' 'Jaccard,' 'Cosine Similarity,' and 'TF-IDF,' respectively. A brief description of each is given in that section.

2.3.1 Edit-Distance

Edit distance is a method used in natural language processing fields for searching, spelling check, etc. through string comparison. I will introduce the Damerau-Levenshtein Distance that is one of the representative methods with an example as shown in Figure 2.11.

Most edit distance methods work with the removal, insertion, and substitution of characters in a string. A method of obtaining the edit distance through an example will be briefly described. It will be found the edit distance between the two strings "RELEVANT" and "ELEPHANT" in the figure. In the first step, "R" should be inserted into the comparison string (ELEPHANT → RELEPHANT). At this time, the editing distance is increased by one. In the next step, replace "P" with "V" (RELEPHANT → RELEVHANT). After this process, edit distance 1 increases further. Finally, remove "H" (RELEVHANT → RELEVANT). This shows that the edit distance is increased 1 and the editing distance between ELEPHANT and RELEVANT is 3. The edit distance also applies to uppercase and lowercase characters. For example, the edit distance of Relevant and relevant becomes 1, because "R" must be replaced with "r."

		E	L	E	P	H	A	N	T
	0	1	2	3	4	5	6	7	8
R	1	1	2	3	4	5	6	7	8
E	2	1	2	2	3	4	5	6	7
L	3	2	1	2	3	4	5	6	7
E	4	3	2	1	2	3	4	5	6
V	5	4	3	2	2	3	4	5	6
A	6	5	4	3	3	3	3	4	5
N	7	6	5	4	4	4	4	3	4
T	8	7	6	5	5	5	5	4	3

Figure 2.11 An example of edit-distance computation

2.3.2 Edit-Distance for Korean

Most of the existing editing distance algorithms have been studied for the alphabet-based on English-speaking languages, and there has not been much research on edit distance for more complicated languages such as Korean or Chinese characters. The study of Korean edit-distance was done by [8] representatively. Roh at el. [8] proposed two algorithms. One of them, SylED, obtains the Korean edit distance by applying inserts, deletes and edits in syllable units. Another method, PhoED, is to divide Korean syllables into several phonemes and then compute the edit distance for each phoneme unit. That is, it is compared the syllables that are aligned and calculate the cost of insertion, deletion, and substitution that occur in phoneme units. These two algorithms are used in combination with a KorED algorithm. If the syllables are different, α is added, and when the phonemes are different, β is added, so that the edit distance can be obtained even if the string lengths are different from each other. Following shows the formula used by the KorED

Base Case

$SED(\Lambda, B(j)) = j \times \beta \quad for \quad |B| \geq j \geq 0$

$SED(A(i), \Lambda) = i \times \beta \quad for \quad |A| \geq i \geq 0$

General Case

$$SED(A(i), B(j)) = \min(SED(A(i), B(j-1)) + \beta,$$
$$SED(A(i-1), B(j)) + \beta,$$
$$SED(A(i-1), B(j-1)) + \delta_S(A[i], B[j]))$$

algorithm to obtain the edit distance.

- A_t that is one-dimensional array for string A means prefix of A_t that has $A_t(i)$ length.
- The similarity score between two different strings A_t and B_t is defined as the sum of the scores of all character pairs in the layout.
- $SIM(A_t(i), B_t(j))$ means $A_t(i)$ and $B_t(j)$ which is a substring of A_t and B_t.

Additional studies have been conducted to improve the performance of the KorED algorithm [9]. The performance was improved to classify typing errors, similar words, and profanity. The algorithm computes the similarity by transforming a word into a one-dimensional array of phonemes to solve complex problems such that a syllable is divided into two or more syllables, or some phonemes are moved to another syllable. Following shows the improved algorithm.

Base Case

$SED(\lambda, \lambda) = 0$

for $1 \leq |B|$

$SIM(\lambda, B_t(j)) = SIM(\lambda, B_t(j-1)) + S(\lambda, B_t[j])$

for $1 \leq i \leq |A|$

$SIM(A_t(i), \lambda) = SIM(A_t(i-1), \lambda) + S(A_t[j], \lambda)$

2.3.3 Jaccard Similarity Coefficient

As shown in Figure 2.12, the Jaccard similarity is a value obtained by dividing the number of elements of the intersection by the number of union elements when comparing two sets. The formula is as follows. However, if both sets are empty, the value is 1.

$$\text{Jaccard Similarity } \mathrm{J(A,B)} = \frac{|A \cap B|}{|A \cup B|}$$

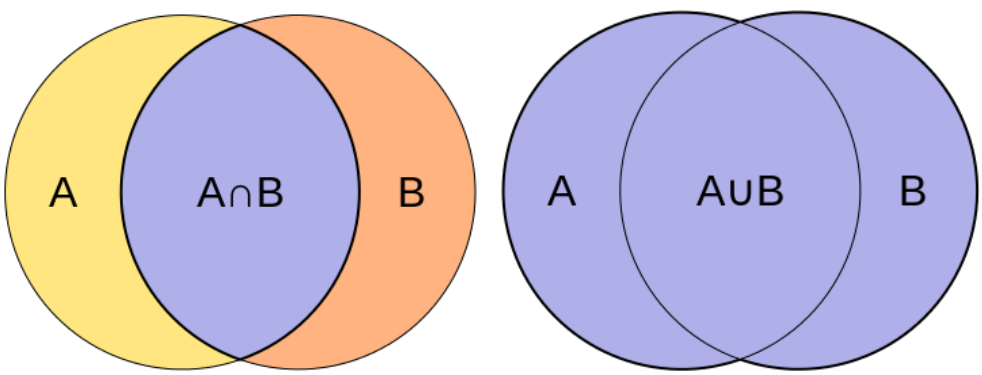


Figure 2.12 Intersection and union of two sets A and B[7]

The similarity is at least 0 to 1 and is shown in the following formula. If there is no intersection, the value is 0. If both sets are equal, the value is 1.

$$0 \leq \mathrm{J(A,B)} \leq 1$$

General Case

$$SIM\big(A_t(i), B_t(j)\big) = \max(SIM\big(A_t(i), B_t(j-1)\big) + S(\lambda, B_t[j]),$$
$$SIM\big(A_t(i-1), B_t(j)\big) + S(A_t[i], \lambda,),$$
$$SIM\big(A_t(i-1), B_t(j-1)\big) + S(A_t[i], B_t[j])$$

[7] https://en.wikipedia.org/wiki/Jaccard_index

This method is used not only for text mining but also for identifying spam mail in social networks by combining with graph theory. In other words, if there are two messages that are highly similar, they are judged as spam.

2.3.4 Cosine Similarity

The cosine similarity refers to the degree of similarity between vectors measured using the cosine of angles between two vectors in the inner space. A typical example is a case of finding similarities between documents. That is, in the field of text mining or information retrieval, each word constitutes a dimension, and a document is a vector value represented by the number of words appearing in the document.

The cosine values of the two vectors are expressed as:

$$\mathrm{a} \cdot \mathrm{b} = \|a\|\|b\|cos\theta$$

Each document corresponds to the words A and B included in the document, and the attribute value is the frequency of the corresponding word.

$$\text{Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

The calculated degree of similarity has a value between -1 and 1, where -1 means completely opposite to each other, and 0 means that they are independent of each other.

The reason why the cosine similarity is widely used is because the Jaccard Coefficient can be applied to the binary data, but the similarity can be measured with respect to the attribute value, not the binary data, except for the attribute matching the zero to zero cosine similarity.

2.3.5 TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is mainly used in information retrieval and text mining. By using this, the importance of the corresponding entity (word) can be known. The reason is that certain words can be expressed as important things in a document. The significance can be found by the occurrence frequency (TF) of the words included in the document. Obviously, the most common words in a document can be perceived as important values, so they can be understood from an intuitive point of view.

IDF is the reciprocal number of document frequency. The total number of documents divided by the number of documents containing the entity (word). This allows you to identify the characteristics of the document set. For example, if the word 'iPhone' appears frequently in a document, this set of documents can be a collection of articles about IT products.

TF-IDF is the product of word frequency and inverse document frequency. The total frequency of word t in document d is called tf(t, d). The formula to obtain tf is as follows:

$$\mathrm{tf(t,d)} = 0.5 + \frac{0.5 \times f(t,d)}{\max\{f(w,d) : w \in d\}}$$

IDF can be obtained by dividing the total number of documents by the number of documents containing the word and then taking the log.

$$\mathrm{idf(t,D)} = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

- $|D|$: Size of Document Set or Counts of whole documents
- $|\{d \in D : t \in d\}|$: Counts of doucumetns including word t

2.4 Information Extraction

Section 2.4 focused on researching information extraction techniques that can be used in extracting the keyword, phrase, sentence from documents. The review aims to survey techniques to construct the knowledge map with the written document.

The aim of Information Extraction (IE) is to extract structured information such as set of entities or relationship between entities from unstructured or semi-structured text by using Natural Language Processing (NLP).

This chapter introduces well-known two type approaches for information extraction.

- **Supervised learning approaches**: Hidden Markov Models, Maximum Entropy Models, Support Vector Machine, Tree-based learning, Condition Random Field

- **Semi-supervised learning approaches**: Corpus, Regular Expression, Mutual Bootstrapping

2.4.1 Hidden Markov Models (HMM) based Approaches

Zhou et al. [10] developed based-HMM chynk tagger in order to recognize and classify names, times, and numerical values on NER. This system can apply and integrate different four sub-features: 1) simple deterministic internal feature of the words 2) internal semantic feature of important triggers 3) internal gazetteer feature 4) external macro context feature. It outperforms any machine-learning and rule-based system.

Skounakis et al. [11] presented Hierarchical Hidden Markov Models for Information Extraction in order to represent the grammatical structure of the sentences being processed.

Using a shallow parser, multi-level-representation of each sentence processed is created, and after that hierarchical HMMs is trained to acquire the regularities of the parses for processing sentences.

Guohong et al. [12] applied lexicalized HMM-based approach to Chinese NER in order to deal with the problem of unknown words. The approach is to unify unknown word identification and NER tagged as a sequence of known words. This system consists of two major modules. The first is known word diagrams, and the second is tagger based on lexicalized HMMs. It has ability to integrate both the internal entity formation pattern and surrounding contextual clues. It achieved outperformance than standard HMM.

Ponomareva et al. [13] considered only POS tag to solve problem non-uniform distribution among biomedical entities. Using POS tags, this system achieves the possibility of splitting the most numerous categories. Although knowledge is not sufficient, this system outperforms than state-of-the-art systems.

Todorovic et al. [14] applied HMM in order to understand low level semantics of texts. This approach is to integrate named entities which are PERSON, ORGANIZATION and LOCATION with grammar based items as DATE, TIME, MONEY and PERCENT.

Jin et al. [15] proposed to extract of customer review and particular product related entities on reviewer's opinion. Opinion expressions and sentences are also recognized and opinion orientations for each known product entity are classified as positive or negative. These approaches have different from the previous works that have mostly based on natural language processing techniques or statistic information. The result shows high performance in web opinion mining and extraction from product reviews.

2.4.2 Maximum Entropy Models (ME) based Approaches

Reynar et al. [16] applied ME to identify sentence boundaries which contain punctuations such as ?, and !. They used trainable model and this model is trained by handcrafted rules, lexica, Part-of-speech tags or domain-specific information. In addition, it can be trained on English texts based the Wall Street Journal corpus and Roman-alphabet languages. Although corpus is insufficient, the performance of this system is better than similar researches.

Kambhatla et al. [17] developed method to extract relations in annotated texts. This approach consists of various lexical, syntactic and semantic feature of the text. It is able to annotate large amounts of unlabeled data without parsing of entire data using the very simple lexical feature. In addition, when various combined features are used, this research achieves the best results.

Xinhao et al. [18] proposed an approach based a maximum entropy for a Chinese text domain. The aim is to segment Chinese word using n-gram language model by applying this approach. This system can shift the word segmentation to a classification task. In addition, a combination of post processing and n-gram language model is applied to the system. Experimental results show that the most performance outcomes in MSRA corpus based on bi-gram language model.

Benajiba et al. [19] developed ANERsys applied on maximum entropy and n-gram approaches in order to recognize an Arabic named entity. This system consists of both an ad-hoc method for the specific Arabic language and gazetteers. They developed training model, corpora and gazetteer to train, and they are used to increase the performance of the system. Although this research showed preliminary result, it has the possibility to tackle NER problems for Arabic language.

Konkol et al. [20] applied to maximum entropy in order to recognize a named entity in Czech language. This approach consists of both the knowledge and experiences acquired from another language. The experiment is conducted by using semantic spaces as a feature for

this approach. It showed the performance than the previous researches in Czech language and provides meaningful approaches for this domain.

Ahmed et al. [21] developed a method based on the maximum entropy for recognizing the named-entity. The aim of this system is to extract the entity sets, which have names, location, and organization of text using the gazetteer list. The maximum entropy among entities is extracted and it is used to train. This approach is confirmed to be the fastest method for retrieving and classifying the entity sets from the database. This system outperforms than the existing related researches for extracting the information.

2.4.3 Function-based Learning (Support Vector Machines (SVM)) based Approaches

Yi et al. [22] introduced a method-applied combination of SVM and an edit-distance. The edit-distance is used to extract the features of SVM. In addition, they proposed a virtual example concept to automatically increase the amount of annotated corpus. This approach is able to resolve the lack of corpus problem. The purpose of the approach is to tackle the problems of spelling variations. The experimental result showed that because of using the virtual example, this system outperforms than other approaches.

Li et al. [23] developed a system uses a feature of the SVM that has uneven margin parameters, and it supports in case of the little training dataset. This approach helps to improve a categorizing performance of document. Through compilation with other state-of-art learning algorithms, the result showed that this approach increases the size of training data and achieved more improvements than other researches.

Aramaki et al. [24] proposed a method including three features: (1) basic pattern feature as lexical patterns (2) selected pattern feature used specific queries (3) physical size feature that is

entity's size. The aim of the approaches is to classify semantic relations by acquiring the size of an entity. The experimental results show this approach is practicable and overcome the problem that other methods do not solve.

Benajiba et al. [25] introduced a SVM based NER method in Arabic language domain. This approach used both specific features for Arabic language and independent language in this framework. This system consists of various features such as contextual, lexical, gazetteers, morphological, nationality and Corresponding English Capitalization. They experimented in both cases that are feature combination and separation. Overall, it achieved high performance as an F1 score of 82.17.

Habib et al. [26] developed an NER system that is based on multi-class SVM approaches and high-dimensional features in order to solve the capability problems of NER. This system eliminated domain dependent knowledge and only used machine-learning method. Their research showed the separated domain approach, which successfully identifies and classify named entities. The result achieved improved training time and showed that this approach is the way to enhance the scalability problem.

Li et al. [27] used Conditional Random Fields and Support Vector Machines to identify named-entity in a clinical free text. This approach applies to the clinical domain by developing an application. They used a gold standard corpus to evaluate this method and selected various features such as dictionary-based, bag of words, POS tags, window size, orientation and capitalization to improve the CRF model as well as SVM's Overall the experiment result showed that CRF outperforms SVM in F-score evaluation.

Singh et al. [28] introduced a Manipuri NER system for an Indian language domain. In order to recognize named entities of Indian text, they applied to two methods with the system. One is an active learning technique using lexical context patterns generated an unlabeled corpus that is annotated with major NE tags, person's name, location name, organization name and miscellaneous name, another one is a method based SVM which classifies different contextual

information in the corpus. SVM used lexical context patterns to improve performance and the experimental results showed high performance for the proposed approach.

Cai et al. [29] developed a system that consists of CRF and SVM and showed its application in order to identify semantic entity. For this system, various features such as context and linguistic and statistical is retrieved using CRF and large-scale text document analysis and used. By combining all features, the SVM conducted classification. Because this approach used the integration in the context, linguistic and statistical feature, the results showed better performance than other approaches.

Habib et al. [30] researched the Named Entity Recognition method based on SVM in the biomedical domain. This is to tackle scalability problems. The approaches are to achieve better performance by eliminating prior knowledge or domain dependent knowledge. The methods both binary and multi-class SVM using increasing training data and are compared with the proposed method. The experimental result showed multi-class SVM reduces training time and the proposed method is more feasible for solving practical environment issues with large datasets.

Ju et al. [31] introduced the NER system using SVM for untrusted biomedical documents. The aim of this research is to identify specifically the name from biomedical texts in large databases. The F-measure (80%) recognized named entity in the biomedical domain is lower than one of a general domain. To overcome low performance, they used SVM and this approach outperforms than the existing methods of the medical domain. The experimental result achieved precision rate = 84.24% and recall rate = 80.76%.

Bj¨orne et al. [32] explored in research for the detection of drug names and statements of drug-drug interaction (DDI) from documents. They developed the system for retrieving Truky Event based SVM for solving two tasks. This system is evaluated by testing three feature sets such as DrugBank, Marama and Both DrugBank and MetaMap. To do this, a semantic parser is intensively used. The experimental result showed this system achieves F-scores of almost 60% for the drug name identification.

2.4.4 Tree-based Learning based Approaches

Paliouras et al [33] used a decision tree induction to optimize a named-entity recognition and classification (NERC) system for a specific domain. This system tags named entities such as persons, locations and organizations and to identify entities uses two resources as both a recognition grammar and lexicon. This approach applied C4.5 decision tree to construct grammars and tested to recognize person and organization names. The results showed that this system outperforms a grammar constructed manually.

Isozaki et al. [34] introduced a method to recognize Japanese Named Entity. In Japanese language, generally approaches based on a Maximum Entropy (ME) showed good performance than decision tree system and handcrafted system. However, systems based ME required much data to train. Due to these issues, an alternative method is proposed by combining a simple rule generator with decision tree learning. The test result showed this system is efficient and appropriate for training with large-scale data set and improves readability.

Black et al. [35] aims to solve name entity classification issues in not English language domain. This system consists of two modules such as the modified Transformation-Based approach and Decision Tree Induction approach to tackle the Part-of-Speech tagging problem. To evaluate this system, they used Spanish and Dutch training data sets and results showed well working Spanish test text.

Because lexical taxonomies have a feature that looks as tree structures, Witschel et al. [36] applied a decision tree to expend lexical taxonomies and his approach is to recognize new concepts and employ co-occurrence entity by calculating relation between words from large scale corpora. These similarities are constructed and insert to each node of the decision tree. The test results showed the overall classification accuracy still low because that lexical taxonomies automatically extended by system itself is until very difficult problems.

Szarvas et al. [37] developed a nation based statistical modelling method. The aim of this

system is to recognize and classify named entities in the Hungarian and English language by using both AdaBoost M1 and C4.5 decision tree as a classifier for learning. In addition, problem-specific methods such as a large feature set, post processing is also applied to this system for supporting it. This research has advantages to be able to apply to different languages without modifying models.

Zhou et al. [38] introduced a medical information extraction system in order to extract meaningful information such as that patient information has breast complaints from clinical medical records. To construct this system, ID3 decision tree techniques are applied to it. The major tasks of this system are to extract medical terms, relations between terms and classify text. Theirs graph-based approach using link-grammar parser to extract relations achieved high performance.

Chakaravarthy et al. [39] developed a system for recognizing the entity from giving relational table. Input data are specific mufti valued attributes and probability distribution for attributes that means the like hood of the occurrence of each entity from the table. This research, unlike previous works concerns the general problems containing inconsistent attributes and inconsistent probability distribution over the set of entities. To tackle these problems, a natural greedy algorithm is applied to the system.

Abdallah et al. [40] introduced an integrated method for a decision tree as a machine learning with a rule based system. The aim of this research is to tackle issues in Arabic language. Their experimental results of hybrid approach results showed that the improvement for the F-measure is up to 8~14% when compared with a pure machine learning system and a rule base system. In addition, it outperforms than the state-of-the-art machine learning systems based a conditional random field.

Oudah et al. [41] aims to develop an Arabic Named Entity Recognition system. In addition, pipelined process is used to solve NER issues. Their hybrid method has the ability to identify 11 different types of a named entity: Person, Location, Organization, Date, Time, Price,

Measurement, Percent, Phone Number, ISBN and File Name. The experiment was conducted in comparison with the three ML classifier and it showed outperformance than the pure ML and rule-based approaches.

Most recent researches for NER is possible to tackle entity recognition in case of specific documents. Because of these problems, Prokofyev et al. [42] introduced a NER approach for characteristic documents such as scientific articles. This approach is composed of a decision tree to classification and n-grams inspection. They evaluate various entity recognition features on a set of computer science and physics papers, and the experimental result showed higher accuracy than other approaches based on maximum entropy.

2.4.5 Conditional Random Fields (CRF) based Approaches

McDonald et al. [43] developed a framework based on a Condition Random Fields for probability sequence tagging in biomedical text. The aim of this system is to construct a model to extract gene and protein mentions in the text. In addition, this approach is able to extend to various biotical entities. The experimental result showed that proposed CRF models using probability tag and lexicons have abilities to recognize these entities without domain knowledge and achieved high accuracy by applying orthographic features and expert features.

Okanohara et al. [44] developed a single probability model based on semi-CRFs for biomedical text. This system has abilities that can tackle long named entities and many labels increasing the computational cost. In order to resolve these problems, they use feature forests to package feature-equivalent state and a filtering process to decrease the number of candidates. Their results showed this system work well by proposing methods without decreasing overall performance.

Peng et al. [45] introduced an application applied CRFs to extract information of scientific

documents such as research papers. This approach is to retrieve various contents such as title, author, and keyword and so on that is paper's information. In order to extract these contents, they defined feature categories that are local, layout, external lexicon features based on the formations of documents. To evaluate they compared various machine-learning methods SVM, HMM. The result showed superior to test approaches.

Zhao et al. [46] applied to CRF in order to resolve Chinese word segmentation such as a character based tagging problem. Although existing researches concerned only feature template, they proposed two methods; a feature selection and a tag set selection. They used and compassed various tag sets to evaluate proposed an approach. The experimental result outperformed than existing approaches.

Bundschus et al. [47] focused on relation extraction problems from biomedical literature. In order to retrieve both relations between entities and type of relations they used CRFs to construct the framework. CRF is very useful to train without feature selection. Their approach is different to previous methods that only concerned detection of relations. The proposed method is able to contribute work of named entity extraction and showed high performance for relation extraction. In addition, it can use general purpose.

Sobhana et al. [48] aims to extract named entities from Geological text. Target texts are scientific documents and articles on the Indian and these are used to construct a corpus. For recognizing various named entity classes, they used various features in context information of words, e.g. current word, POS information, digit features and so on. To evaluate this system, they compared with other methods such as SVM. The experimental results showed higher accuracy than other methods.

Rocktäschel et al. [49] proposed applied a CRF trained from chemical text that are consist of chemical entity such as trivial names, drugs, abbreviations and so on. In order to extract entity, they used a combination CRF with a dictionary. CRF is appropriately used to extract morphological entities and a dictionary is used to extract short and inappropriate named entities.

Due to the advantages of this approach, they showed the proposed system outperforms existing methods and is possible to be used for a broad chemical entity recognition.

2.4.6 Regular Expression based Approaches

Li et al. [50] presented a method to reduce manual effort such as creating high quality and complicated regular expression for information extraction process. They developed RELIE, transformation-based algorithm for learning such as regular expressions. In order to evaluate this algorithm. They compared with CRF algorithm and showed RELIE is faster than CRF and outperform CRF under training shortage of data and cross-domain data. Totally. They showed which CRF's performance can be improved when extracted features by RELIE used.

Brauer et al. [51] introduced a method automatically reasoning regular expression from sample entities such as retrieved from a database or an annotated document corpus. The proposed approach can learn efficient regular expressions that can be easily understood by a user without any document corpus and extension of the application of information extraction. From weighing dependent entity features, this system can achieve high performance by selecting the most suitable regular expression form.

Eka et al. [52] developed a system for short text messages in Swedish written in a mobile environment. This system retrieves locations, names, times, and telephone numbers because these entities can be used in other applications. In order to implement this system, a regular expression is applied to support a classifier based on logistic regression. This approach showed fast response time on the telephone and achieve high performance F-score of 86.

He et al. [53] present an approach for a web forum. A main task of this system is to learn the specific features of forum systems and decide how to select the suitable features to construct the system fingerprints. The system has two main tasks: 1) by clustering those pages, page layouts

and contents are identified 2) the features of the forum system are required and these can be used to extract information. The experimental results show this system well work for user's information extraction.

Sawsaa et al. [54] proposed Java Annotation Patterns Engine (JAPE) based on the Information Science concept to construct a domain ontology. The JAPE supports a regular expression matching. This can reduce the consuming of ontology construction time. The experimental result showed the pattern matching used lookup list produced 403 correct concepts and missing and the false positive result are not. Thus, the proposed method is the best approach to help expert's work and efficiency of work.

2.4.7 Corpus based Approaches

Tanabe et al. [55] proposed a tagged corpus for gene/protein named entity recognition in biomedical texts. The purpose of the proposed system is to reduce a difficulty of extracting gene/protein names due to the complexity of gene/protein names. In order to resolve these problems, they developed annotation of GENETAG; a corpus has 20k medicine sentences for gene/protein. Although annotators are required to judge the annotation of GENETAG and the data have to pre-parse to word. They showed character-based indices outperforms word-based indices.

Szarvas et al. [56] developed a method to increase accuracy of named entity corpus for Hungarian. In order to create a corpus, a parallel annotation process conducted by two annotators and the result showed a tagging with inter-annotator agreement rate of 99.89%. To increase reliability of the corpus, two annotators have discussed with a linguist with experiments for several years. To evaluate the corpus, various machine learning algorithms and classifiers are used. The experimental result showed high accuracy of 92.86% F measure on the corpus.

Pyysalo et al. [57] introduced a BioInfer (Bio Information Extraction Resource), a corpus of

biomedical English. The aim of research is to acquire protein, gene, and RNA relationship from biomedical texts. In order to construct a corpus, they analyzed an annotation scheme for named entities and their relationship based on sentence syntax. Moreover, they created the ontology which consists of a type of entitles and relationship in the corpus. Using 1100 sentences of biomedical literature, the corpus is constructed.

Rosenfeld et al. [58] introduced methods to increase Web Relation Extraction (RE) using corpus statistics. In order to improve RE, they presented a method to use corpus statistics, validate, and correct the issues of extracting relation. To evaluate this approach, they used the method based on a self-supervised web relation extraction system and compared with both simple rule-based NER and a statistical CRF-based NER. They showed high performance for relation extraction and validation.

Tomanek et al. [59] presented the Active Learning (AL) for the annotation of named entities. This approach showed faster annotation work under real-environments. In order to construct a corpus, they used specific classifier and feature sets. They showed the rate of reduction for annotation efforts until 72%. The experimental result showed AL can be used for various applications, but, to construct a corpus, large amount efforts for annotation are required.

Ekbal et al. [60] developed a tagged Bengali news corpus using web resources of Bengali newspaper. Using web crawler, the web pages in HyperText Mark-up Language extract from news resources. The corpus has approximately 34 million words at present. Their NER system used pattern based shallow parsing and linguistic knowledge created by this corpus. The experimental results showed high accuracy, overall F-score of more 70% for a person, location, organization and miscellaneous names.

Kipper-Schuler et al. [61] introduced Mayo Clinic Information Extraction system to extract the named entity disease. In order to construct this system, a corpus, which has 160 free-text clinical literature based on manual annotation, is developed. To evaluate this corpus, they use a subset SNOMED-CT with semantic types of disease and disorder mentions. They archived F-

score 56% in the case of exact matches and 76% of right partial matching and 62% left-partial matching.

Roberts et al. [62] introduced the building of a corpus in which clinical texts contain both multiple entities and their relationship. In order to construct this corpus for a CLEF project, they used large corpus, handled multiple text types such as clinical narratives, radiology reports and histopathology reports and over 20 annotators have worked. This corpus consists of both structured records and free text literatures from hospital for deceased cancer patients.

Ohta et al. [63] used GENIA as a corpus and GENETAG in order to extract proteins and genes in molecular biology. The GENETAG conducts an annotation for the conceptual entity, gene and GENIA is used to recognize forms of gene, protein, DNA and RNA. These features can solve various problems such as the compatibility and comparability of the annotations. In this research, they showed a combination method which an extension of GENIA integrated with GENETAG gene annotation.

Desmet et al. [64] proposed a named entity corpus for Dutch. Although existing named entity recognition system uses large annotated corpus in English domain, there is not in Dutch. The aim of this research is to construct a corpus of a various 500-million-word containing named entities, co-reference relations, semantic roles and spatiotemporal expression. In order to construct this system, they trained 1-million-word sub-corpus and developed automatic classifier.

2.4.8 Mutual Bootstrapping based Approaches

Lee et al. [65] developed a bootstrapping method for applying to geographic named entity annotation. In order to construct a system, they build a raw corpus by annotating with seeds. Using annotating, they trained boundary patterns and these applied to the corpus as new candidates. To reduce over a generation, type verification is used. They showed the bootstrapping

method provide increasing annotated instances and becoming boundary patterns richer.

Kozareva [66] proposed a method, which automatically generates a gazetteer list from unlabeled data and construct named entity recognition system using labelled and unlabeled data. The aim of this research is to solve problems of NER, which lack of the handcrafted data and low scalability in another domain. Through unlabeled data, they easily construct the gazetteer list for the person and the location and the gazetteer is used as a feature for named entity recognition system.

Pennacchiotti et al [67] proposed a combination method that consists of weakly supervised iterative algorithm and web-based knowledge expansion technique in order to retrieve binary semantic relations. Using small seed instances for specific relation, the system learns lexical pattern and used them to acquire new instances. Through these methods, system can expand the instances. They achieved high performance for extracting various semantic relations than two state-of-the art systems.

Van [68] introduced automatically generating multilingual geographical name gazetteers based on two bootstrapping with different corpora. In order to construct this system, they matched small seed-list of geographical names to an unannotated dataset for one language and used memory-based learning to extend gazetteers. The proposed is similar to co-training technique.

Arguello et al. [69] presented a method for identifying stakeholder mentions in natural language text. In order to construct a system, they used a bootstrapping technique and categorized stakeholders into two types, which are he and she. Their bootstrapping is combined with three different extraction pattern templates. The experimental results showed that the proposed method can be learned using small extraction patterns and is suitable to identify stakeholders.

Lee and Lee [70] developed a geographic named entity recognizer based on a bootstrapping algorithm with error correction methods and location normalization. Though location normalization, ambiguities of entities can be resolved. In addition, a corpus is created by

annotating with a large set of seeds. Bootstrapping algorithm help annotated instance, gradually increase and learns boundary patterns to improve the system performance. They achieved high performance, 89 of F-measure.

Dang and Aizawa [71] presented the use of dependency trees in order to extract patterns and relation for entities. They used a bootstrapping technique to improve the performance of a NER system and to compute tree patterns. In addition, they developed the technique of simultaneous multi-class bootstrapping, which highly improve the quality of the seeds.

Kawai et al. [72] introduced a cost-effective web search (CESS) framework in order to retrieve keywords in the semantic class of Web. To construct a corpus, a bootstrapping technique is applied to the proposed system. In addition, in order to gather knowledge for recognition open web API is used and this provides powerful performance. However, there are limits for using API calls; thus, they proposed optimized web search method using least API calls. They can retrieve 64,642 words for 5 different domains and achieved precision of 0.94.

Venturi et al [73] developed a large-scale lexical resource for the biomedical domain in order to extract domain specific information. A bootstrapping technique is used to construct a biomedical corpus. This corpus contains verbs, semantic event frames and automatically extracted syntactic frames. In addition, the corpus includes manually added links between semantic and syntactic. They showed their corpus is unique resources for the biomedical domain.

Wu et al. [74] introduced a domain adaptive bootstrapping (DAB) method in order to resolve domain adaptation problems. The bootstrapping is conducted using trained classifier for labelling unlabeled data. This approach is useful when labelled training data is low and unlabeled data is large. The experimental result showed their method outperform supervised methods and other methods applied bootstrapping.

Polifroni et al. [75] developed a dataset and named entity recognition system based on a bootstrapping method for mobile services. They create a corpus using a very large dataset and

classifier to identify named entities in speech of mobile. Their contribution is to construct the process of creating the data and selecting the set of features to named entity recognition by using simulated data and large knowledge resources for related data.

Glass and Barker [76] presented a bootstrapping technique to extract relation between entities. Because a pair of words can have semantic relations in sentences, the relation extraction is a very meaningful task. They conducted a training for extracting relation using relation extractor and tested to verify relationships in two domains using non-parallel and parallel corpus composed of new articles. They showed an approach used parallel corpus greater than other corpus.

Putthividhya et al. [77] developed named entity recognizer in order to extract product attributes and values from listing titles. To construct a NER system, they used both supervised NER and a bootstrapping technique. They focused on listing form of eBay's clothing and shoes, categories. Their bootstrapping can recognize new brands and detect spelling errors for known brands. They achieved high accuracy of 90.33% precision.

Schone et al. [78] presented a near-zero-cost methodology to construct without significant human efforts and computational cost. Their approach is to build a relation extractor in not English for Wikipedia and other web pages and other English knowledge. They applied this method to Greek, Spanish, Russian and Chinese. They evaluated performance at the file level and achieved high accuracy.

Sun and Grishman [79] introduced a general cross-domain bootstrapping algorithm for identifying named entity. They conducted a task to generalize the lexical features of domain model using clustering by a joint corpus. The bootstrapping conduct selecting target domain instances. Their approach can achieve high accuracy 70% of F-measure, without annotated data and without knowledge of target-domain.

Teixeira et al. [80] applied to a bootstrapping algorithm for training in order to construct an

NER system. They used almost 50,000 annotated people's names, and this is used to conduct a simple corpus based approach. Using these training sets, they construct a classifier based on Conditional Random Field. This classifier is used to build additional annotation to a corpus. In experiments, CRF conduct seven iterations and they achieved 83% precision and 68% recall.

2.5 Plant maintenance

Section 2.5 reviews the techniques and applications that have developed for maintaining large industry plants. The section includes the overall trend of an industrial plant management process, and the review of the purpose, method, and outcome.

2.5.1 Failure Detection

Sensor networks and alarm has been used in order to detect and predict functional failure in the large industrial plant. The most popular approach in failure detection is the simple outlier detection from alarm and sensor data. The detected outliers, unusual patterns in the alarm or sensor data, do not always end up with the severe issue or failure, as well as to require human expertise in order to define the type of failure. Hence, it is almost impossible for human expert to diagnose the failure and provide the appropriate solution in a short period of time [1, 4, 81].

Foong et al. [2] aims to prioritize the alarms during alarm floods which would ease the burden of operators with meaningless or false alarms by using fuzzy logic and 125 fuzzy rules. To facilitate in rule construction, five linguistic values are used to determine the ranges of the criticality for each parameter which are lowlow, low, normal, high and highhigh. These ranges of values are gathered from oil refinery engineers or experts. For the output, four different categories of alarm prioritization are used which are 1) normal, 2) low, 3) high and 4) emergency.

Nan et al. [82] proposes a knowledge-based fault diagnosis method using the worthy knowledge from the experts and operators, as well as real time data from various sensors. The Methods used were Fuzzy logic and five output functions.

Abele et al. [83] developed an alarm system that performs Root Cause Analysis (RCA) upon an alarm model constructed with Bayesian networks. In the paper, methods are presented to

construct Bayesian networks for RCA (Root Cause Analysis) with a knowledge-based and a machine learning approach.

Aizpurua et al. [84] aims to build a rule based expert system is used to find the “Alarm Root Cause”. The system finds the root cause of avalanches of alarms and their effects and reduce their number through grouping or clustering techniques, complying with the EEMUA 191 standards.

Zhao et al. [85] proposed a power system alarm processing and fault diagnosis expert system (AFDES). In the proposed expert system, Backus-Naur Form (BNF) is applied to design a type of expert rule frame which operator can write and increase the rules with his own defining language to rule-base.

Ebersbach and Peng [86] developed the first artificially intelligent system for fault diagnosis and machine condition monitoring which use integrated analysis of vibration, oil and wear debris analysis technique. It designed and implemented an expert system for analyse vibration data with similar accuracy as an maintenance engineer in an automated software package allowing high analysis performance, and hence suitable for commercial machine condition monitoring laboratories or fields use.

Safavian and Landgrebe [87] aims to reduce the number of alerts presented to the operator. It used a rule-based method. 6 knowledge bases are built, and the rules describe typical interrelations between alarm messages which have a common cause. The concept has been implemented in a software prototype which manages the alarm log, plant model and interrelation rules and presents the grouped alarms in an interactive alarm display. The alarms are not deleted from the alarm logs. Rather, it is the same alarm log but structured hierarchically. The result is a compact alarm display with fewer alarm messages visible on the top level but a higher information density. The application of the approach on two case studies resulted in a successful reduction of alarms.

Folmer and Vogel-Heuser [88] presents an overview of an algorithm for the automatic alarm

data analyzer (AADA). It is able to find possible and significant reasons for alarm floods by identifying the most frequent alarms and those causal alarms consolidating alarm-sequences. 12.000.000 alarms are used as a dataset. For the experiment, the alarm logs have been available from four different industrial process control and manufacturing plants as case studies, e.g. purification plant (continuous process), hydraulic fiber press (discrete and continuous (hybrid) process) and incineration plant (continuous). The results demonstrated from AADA (the automatic alarm data analyzer) that this data can be used for the redesign of an AMS (alarm management systems) to reduce alarm floods and reduce operator's workload.

Ahmed et al. [89] proposed an alarm system framework with various types of alarm data management system, including data filtering system, alarm delay, and alarm deadline settings.

Izadi et al. [90] described and evaluated the most efficient alarm filtering system. They presented the alarm filtering approach that calculates the similarity in the alarm and sensor data sequence, and clusters them in each group. The research conducted by [91] was focused on finding an alarm flooding management system. They proposed a dynamic alarm management approach by applying the Bayesian Network technique.

The above researches aimed at analysing the characteristics of alarm data, and managing the size of an alarm. This would limit the participation of domain experts, and most of the researches are not evaluated to check whether the detected alarm or sensor data pattern affects the real failure.

2.5.2 State-of-the-Art on Knowledge Engineering Techniques for Failure Diagnosis

Several machine learning and data mining techniques were applied for industrial failure diagnosis. Yin et al. [92] introduced machine learning-based online fault diagnosis by using incremental support vector data description (ISVDD) and extreme learning machine with

incremental output structure (IOELM). An online fault diagnosis approach combining ISVDD and IOELM could detect new failure mode and recognise fault based on learning knowledge of the diagnosis system.

Extreme learning machine(ELM)-based real-time fault diagnostic system for gas turbine generator systems was proposed [93], and compared with the most successful machine learning algorithm, including support vector machine. The evaluation result is 98.22% accuracy in 2.7 ms. The proposed ELM fault diagnostic framework is generic; it could be applied to the other applications of condition monitoring in which the fault identification time is critical.

Wind turbine failure diagnosis system applied a binary tree SVM and a self-organising feature map neural network [94]. Fuzzy logic, support vector machine (SVM) and artificial neural networks were employed for continuous monitoring and fault diagnosis for monoblock centrifugal [95]. Feature extraction using wavelets and SVM algorithm for classification are successful approaches for practical applications in industrial fault diagnosis.

Li and Zhao [96] proposed gravitational search algorithm (GSA) to identify and diagnose new fault samples by calculating the weighted kernel distance between them and the fault cluster centers. The proposed method has been applied in unknown fault diagnosis, and evaluation results have shown the effectiveness of the proposed method in achieving expected diagnosis accuracy for both known and unknown faults of rotatory bearing. The application of evolving fuzzy modeling to fault-tolerant control was proposed in two steps: fault detection by applying model-based approaches and fault accommodation by using fuzzy models [97].

2.6 Ontology Engineering

Section 2.6 finally reviews the ontology engineering approach that can be used in building the knowledge map. The proposed knowledge map does not directly follow the ontology development method but the major idea was coming from its engineering approach.

2.6.1 An Analysis of Ontology Engineering Methodologies

Ontology's critical role in machine understandable web is now widely accepted. Therefore, many methodologies have been proposed over the past two decades but still this field lacks mature and widely accepted methodologies [98]. This is happening because most methodologies do not offer enough details of the techniques and activities that they use [99]. One of the main reasons also includes most methodologies being applied for developing ontology for a specific project. Terms and definitions relevant to business enterprises are collectively called the enterprise ontology, on which the methodology developed by Uschold and King was based on [98]. They were the first ones to propose a methodology for developing ontologies [100]. However, they do not specifically describe the techniques and activities [101]. Gruninger and fox's proposal was similar, their methodology also relate to the business domain, which was derived from the experience of creating the TROVE project ontology [99]. Informal intended semantics were captured from motivation scenarios that evolve the competency questions for the ontology to answer [102]. The activities and techniques here is also insufficiently detained [102]. Methontology, the methodology that contains the detailed description of activities and techniques, was introduced to serve the purpose of building domain ontology from scratch [99]. Methontology creates knowledge level and has a life cycle based on evolved prototypes [103]. It includes development-oriented activities like specification, conceptualisation, formalisation, integration, implementation and support activities like knowledge acquisition, evaluation, integration and documentation [103]. It supports the notion of reusability [98] and have been used in the domains of chemicals [99], environmental pollutants [104], monatomic ions, silicate ontology [103] and

many others [105]. IDEF5 has an evolving prototype model and is application dependent. It is mainly used for the communication between the domain expert and ontology developer whose initial response is later transformed into a structured language based on KIF [106]. The library of IDEF5 contains definitions and characterisations of commonly used relations [106]. However, it does not specify a life cycle and only provides limited details [107]. SENSUS was developed using various sources of knowledge and some electronic dictionaries [98]. All irrelevant terms are pruned in the final ontology of SENSUS [110]. Like many other methodologies, SENSUS does not mention any particular technique or detailed description [99]. The CYC methodology, which has three phases, is developed to represent common sense and encyclopaedic knowledge [99]. Mikrocosmos is a project developed for machine translation, which contains general development guidelines and useful heuristics [109]. The Plinius project is another project similar to Mikrocosmos, which also includes general guidelines applicable to other domains, but they may not always be enough to cater to them [111]. Ontologies play a vital role in common KADs which is a widely used methodology for developing knowledge base systems [107]. On-To-Knowledge methodology handles enterprise solution by intending to bring a balance between human problem solving and automated IT solutions [112]. Some methodologies [113] including ONION [114] and MENELAS [115] work with medical domains. One of the presumptions of MELENLAS is idealised view of taxonomies that makes it incompatible with other domains. 101 method explains and elaborates the purposes of authors by using a wine ontology as an example [107]. UPON [116] adopts UP and UML which makes it handier for both domain experts and knowledge engineers [117].

Some methodologies that provide details including Methontology can be compared with one another based on criterion, trends and needs. Some methodologies do not pay equal attention to all the aspects of ontology. For example, many methodologies that put focus on domain analysis and scope identification may lack due attention in the design phase. A criterion is defined using eight different aspects that contributes to develop a quick understanding of different methodologies. This will also help in choosing the right methodology for projects with specific needs [98].

Type of development, support for collaborative construction, support for reusability and support for interoperability are the first four aspects of the criterion, which reflect high-level details of a methodology. However, specific and technical details are not discussed in case of these four. Degree of application dependency, life cycle recommendation, strategies for identifying concepts and details of methodology are the aspects that cover the technical side of a methodology [98].

Stage based model, evolving prototype model and guidelines, depending on the type of development model they follow are the three broad categories in which the literature can be divided. Different approaches based on these categories have their own pros and cons. Scenarios where the purpose and requirements are clear are good for stage-based categories. Whereas evolving prototype may be the best choice when requirements are initially unclear and require refinement over time. Recommendation of useful tips, rules and techniques, for making better design decisions rather than focusing on the overall development model are the main focus of guidelines. This distinct classification helps in choosing the best approach available [98].

The construction of ontologies can be isolated as well as can be done in collaboration. Extensive teamwork can be facilitated on the same ontology through Collaborative construction support. In this case, team members are not restricted to a geographical location; contributions can be made from any location without putting any effect on the project efficiency. This is particularly useful for a fast-paced world where people want to work from remote locations [98].

Developing ontology is a very time consuming and tedious task. As reusability eliminates need of repetitive tasks, the idea of reusability of ontology is becoming very popular [98]. Bottleneck ontolingua server was introduced to overcome this, which also contains the feature of collaborative ontology construction [98]. But it lacks details about mapping functions [108]. CYC [109] and SENSUS [110] also supports the notion of reusability. Reusability allows ontology engineers to make use of existing ontologies, which reduces the overall ontology development time and efforts. This also allows other advance issues to be more focused on, like ontology quality. Therefore, supporting reusability is very important for an ontology [98].

Today's ontology engineering includes interoperability as an important aspect. Interoperability between systems is supported by many methodologies. The same skeleton or high-level concepts are used by domain ontologies that use these methodologies so that they can communicate and share knowledge with each other [98].

Application dependency during ontology development is distinct across different methodologies. The three scenarios that can be opted by a methodology include application dependence (ontology is developed on the basis of an application knowledge base in mind), application semi-independence (possible scenarios of ontology use are kept in mind during the specification stage) and application independence (no assumption is made regarding the uses to which the ontology will be put in knowledge-based systems, agents, etc) [98].

The set of stages through which the ontology moves during its life is called a life cycle which is not clearly recommended by many of the methodologies. One of the criteria is having a life cycle [98].

Identification and inclusion of candidate concepts in ontology design is unquestionably an important process. Bottom-up approach, top-down approach and middle-out approach is some of the approaches used by this process. Preference for a specific approach is not unanimous amongst academics. This depends on their experiences and the nature of the project [98].

Some activities and techniques to support ontology development are consistent amongst all methodologies. But the level of details regarding the techniques they employ varies greatly. Methodologies are classified to have three degrees of details including sufficient details, some details and insufficient details [98].

TOVE and Enterprise model approach fall into the same criterion. Both of them stage based type of development, no collaborative construction, life cycle recommendation or interoperability support, have reusability support, middle cut strategy for identifying concepts, some level of

details and are semi-independent. KBSI IDEF5 is the similar but it's strategies for identifying concepts is not clear. Methontology has similar attributes as TOVE but has sufficient details instead. Ontolingua is the same as KBSI IDEF5 but with modular development and collaborative construction. Common CADS and KACTUS has top-down strategy and insufficient details. Plantis follows the guideline type of development and bottom up strategy. Onions have both modular development and guideline with interoperability support and no clear strategy. Mikrocosmos has rule-based strategy and Menelas has concept graphs. Sensus does not mention any preference in strategy and is application semi dependent. Cyc methodology, UPON, 101 method and On-to-knowledge are application dependent. Upon and on-to-knowledge uses middle cut strategy while 101 method uses developer's consent [98].

After comparing with the aspects of the criterion, it can be derived that none of the methodologies are fully mature. Excluding some exceptions (most prominently METHONTOLOGY), most of the methodologies do not provide sufficient details. The notion of reusability is provided by only a few, whereas even fewer methodologies provide details about that. Only Ontolingua is worth mentioning in case of collaborative construction [98]. Conventional strategies for identifying ontology concepts are used by most methodologies. However, the ontology engineers are to explore new ways and techniques to make this process more efficient and easier. [98].

2.6.2 Ontology Engineering and Development Aspects

Ontology, a hierarchical representation of the properties and instances of classes and subclasses, has created key concepts such as domains, derivation of relationships and representation of them in machine interpretable language [118]. Applications of Ontologies derived the realisation of semantic web [119]. Advancing towards semantic web performs intelligent search and stores result in distributed databases [119]. Ontology is the backbone of Semantic web [119]. Defining data on web and linking them in a way that the data can be

understandable by machines is called semantic web [120]. Logical theory contributes towards developers building an explicit and partial account of a conceptualization [121]. Ontologies are sharable and acceptable generic knowledge bases [119]. Different types of ontology include upper ontology, heavy weight, lightweight ontologies, and domain and task ontologies (Mizoguchi, 1995). Standard upper ontology is defined to cover 3D and 4D visualisation aspects of ontologies that have not been reached it [119]. Google, Yahoo and other search engines are covered by lightweight ontologies. Hierarchical structures are created by heavy weight ontologies that has principally philosophical motives [119]. Both domain and task ontologies are used for specifications [119]. Theory of all vocabularies are provided by task ontology while domain ontology is used to define relationship amongst classes [119]. Scope identification, elaborating resources, defining taxonomy, defining properties, defining facets, defining instances and verifying for ambiguities are the steps of building ontologies manually [123]. Ontology editor tools like web protégé, Hozo, Knoodl, vitro etc. are used to improve, reuse and modify existing ontologies [119]. Ontology can remove word sense disambiguation, reuse and analyse domain knowledge, be helpful in solving reasoning problems and help in achieving interoperability in semantic web [119]. Methontology is an example of an ontology with a specified life cycle that can perform development oriented activities and support, management or integral oriented activities [123]. Management activities include scheduling and quality assurance while development oriented activities can be classified as pre-development, development and post development activities [119]. Deciding the type of environment and doing a feasibility study are parts of the pre development activities while development activities include specification, conceptualisation and formalisation phase. Maintenance is one of key post development activities, which require well defined guidelines [123]. Ontologies can be reused in order to formalise knowledge into a form that everyone can understand [119]. Support activities include knowledge acquisition and configuration management [119]. Tools that support ontology engineering have already been specified [124]. Evaluation on ontology evaluation frameworks have also been extensively carried out [125]. One of the elements in KAON ontology is KAON OImodeller [119]. One of the useful ontology-engineering workbench is WebODE [119]. Another open source network that does the integration and management of different aspects of given ontology is OntoEdit [119].

3 Alarm Data Analytics

I focus on detecting failure status by using alarm data in particular on large industrial plant. The section describes how I collected a set of alarm data and classified the facility status based on the alarm data.

3.1 Alarm Data Collection

I use the alarm data that was collected in the industrial plant from the Hyundai Steel Co., Ltd for a 1-year period (from September, 2015 to July, 2016). In the plant, most alarms are connected with one or more sensors to indicate the facility activities, and an alerting device to detect any failure. I collected over a round half million, 567,748 alarm data from an industrial plant. Figure 3.1shows the alarm data collection interface with the detailed information of some example alarm data that was collected on 25th of July 2016.

In the figure, the alarm is integrated and provided from the two different sources, HMI (Human Machine Interface) and CMS (Central Management System). I merged alarm data from those two sources and visualized this to the interface as can be seen in the upper-right corner of the Figure 3.1. When new alarm is occurred, the alarm integration system collects all detailed information of the specific alarm, including starting time, ending time, facility id, alarm message, and ratio (%). For example, the first alarm data in the figure indicates the 'forward press entrance inhibits' issue in the Slab Sizing Press (SSP) Area, which occurred from 01:07am to 03:13am. The last column 'Ratio (%)' describes how much capacity that alarm took so it uses 100% of working memory at that period.

3.2 Alarm Data Feature Analytics

Facility Failure Status Assessment:

Next, I focus on assessing the facility status. The collected alarm data is traditionally sent to the human experts and experts diagnoses and treating the failures. However, the aim of this research is proposing automatic failure detection framework by using machine learning and human expertise. Therefore, it is crucial to have class/label for defining the status of facilities based on alarm data.

For this task, I asked 35 human experts, who have experienced various types of industrial disasters from the industrial plant of Hyundai steel Co., Ltd since they have sufficient knowledge in diagnosing and treating failures by reading and analyzing the alarm data. Based on the focus group with 35 domain experts, I identified 50 different facility statuses that would be used as class labels.

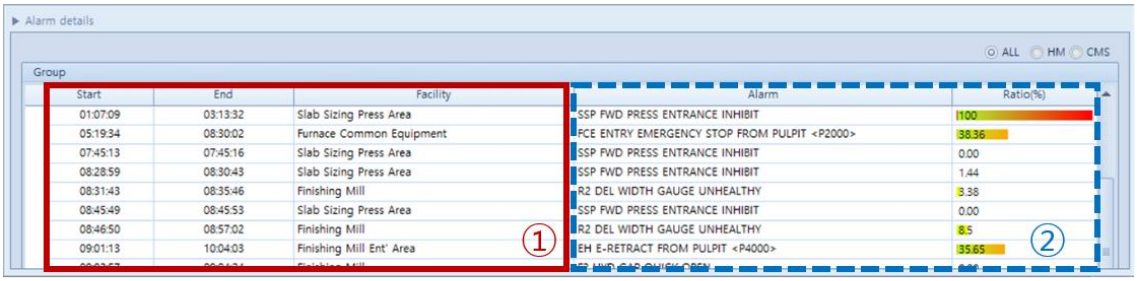
The following are 50 facility statuses identified by domain experts: 2passfault, apc, breakaway, bur, bwd, carbonization, close, collision, corrosion, cradle, cut, damage, defective, division, down, fault, flame, gap, heavyfault, hunting, impact, intrude, leak, nocooling, noenter, nolink, nooff, nooperation, noreversefwd, norupture, nosense, nostop, obstacle, on, open, permeate, plateloadon, position, relaxation, slip, slowincome, speed, stop, transform, trip, up, vibration, wronginputpower, wrongoperation, wrongsense.

Based on the given 50 class attributes, I asked 35 experts to classify and label the status of facility by reading and analyzing the provided alarm data. The labeling procedure is as follows. A class label for an alarm data was assigned if 21 out of 35 experts (60% of experts) agreed on the label. In another case, I selected the first and second rated labeled, and asked experts to choose one of the labels. For example, let's label the first alarm data in figure 2 with 35 experts. 40% of 35 experts labeled it to 'noenter' class and 35% of experts classified it into 'obstacle' class. In

this case, I asked experts to classify the data by picking ‘noenter’ or ‘obstacle’ class. With this procedure, the alarm data is labeled into ‘noenter’ class.

Before performing the failure status detection by using alarm data, I analyze the characteristics of feature values in the training data. To illustrate the discriminative capacity of these features, I deploy box plots for each of them. In this analysis I distinguish it with the range of its attribute value. The box plots are shown in the following Figure 3.2, Figure 3.3, Figure 3.4 and Figure 3.5.

In order to illustrate the nature of time-series alarm data frequency, I plot the real-time alarm data in line graphs. I show these line graphs in the following figures. The Figure 3.6, Figure 3.7 and Figure 3.8 shows each yearly, monthly, weekly alarm frequency. As figures show, it does not have any specific alarm frequency pattern but just consistent circumstance.



①

Start	End	Facility
01:07:09	03:13:32	Slab Sizing Press Area
05:19:34	08:30:02	Furnace Common Equipment
07:45:13	07:45:16	Slab Sizing Press Area
08:28:59	08:30:43	Slab Sizing Press Area
08:31:43	08:35:46	Finishing Mill
08:45:49	08:45:53	Slab Sizing Press Area
08:46:50	08:57:02	Finishing Mill
09:01:13	10:04:03	Finishing Mill Ent' Area

②

Alarm	Ratio(%)
SSP FWD PRESS ENTRANCE INHIBIT	100
FCE ENTRY EMERGENCY STOP FROM PULPIT <P2000>	38.36
SSP FWD PRESS ENTRANCE INHIBIT	0.00
SSP FWD PRESS ENTRANCE INHIBIT	1.44
R2 DEL WIDTH GAUGE UNHEALTHY	3.38
SSP FWD PRESS ENTRANCE INHIBIT	0.00
R2 DEL WIDTH GAUGE UNHEALTHY	8.5
EH E-RETRACT FROM PULPIT <P4000>	35.65

Figure 3.1 Alarm Data Collection Interface

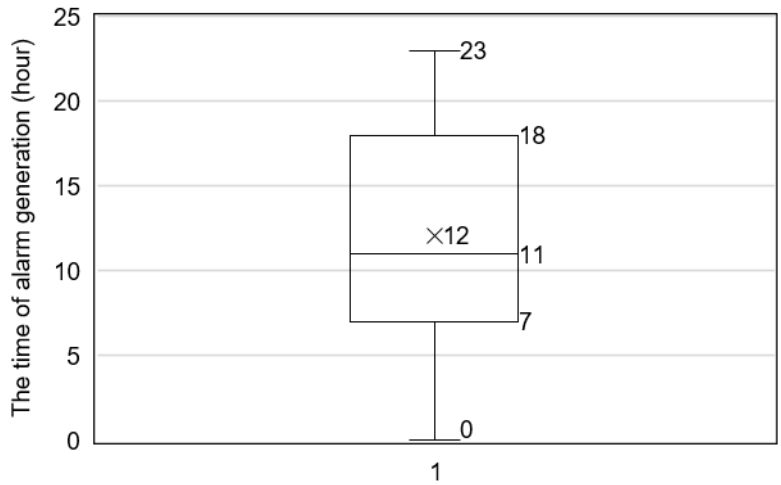


Figure 3.2 A box plot for the average time of alarm occurrence

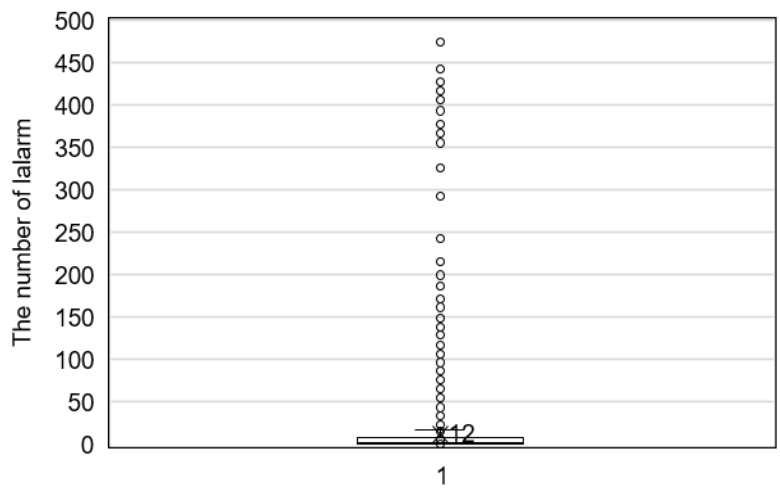


Figure 3.3 A box plot for the average of all alarm occurrence in 1 hour

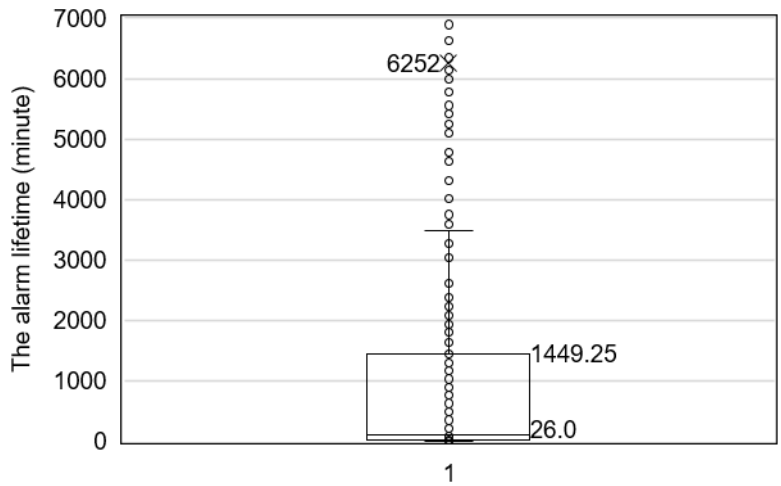


Figure 3.4 A box plot for the average lifetime of all occurred alarms

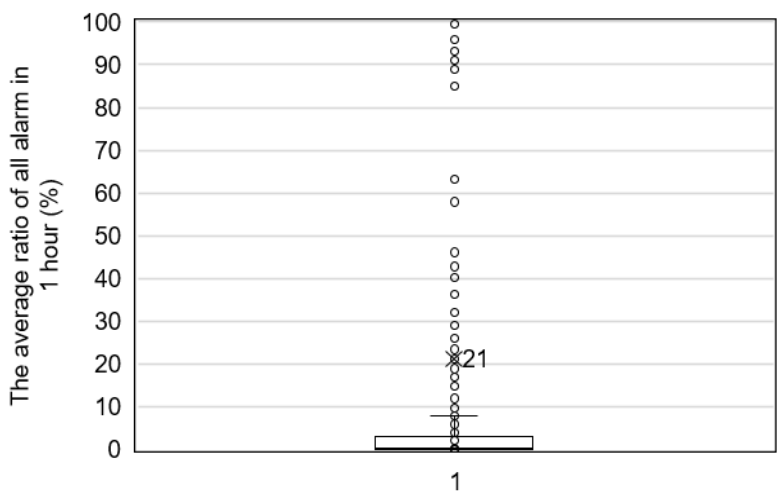


Figure 3.5 A box plot for the average ratio of all alarm in 1 hour

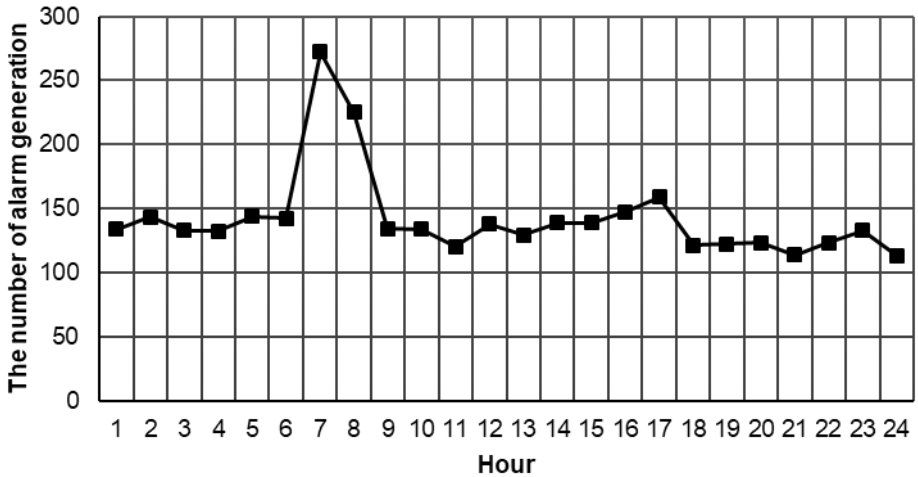


Figure 3.6 The trend of average of alarm occurrence for five months
(1 year from September, 2015 to July 2016)

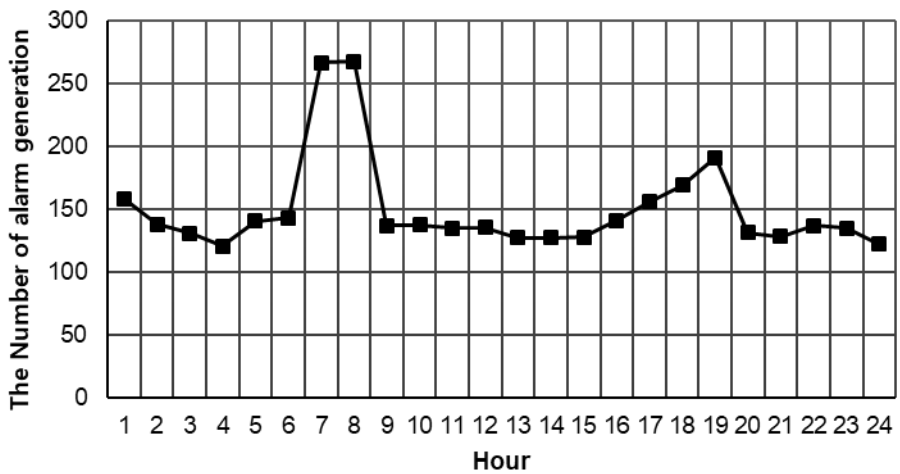


Figure 3.7 The trend of average of alarm occurrence for one month months
(1 month from 1st December, 2015 to 31st December, 2015)

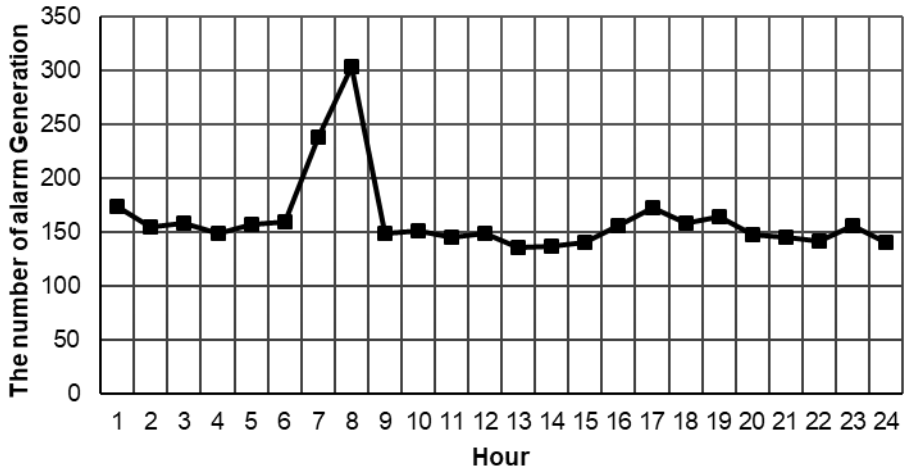


Figure 3.8 The trend of average of alarm occurrence for one week
(1 year from 1st December, 2015 to 7th December, 2015)

Feature Analysis for Failure Status Detection:

I propose a set of features to characterize alarm data in our collections. The features were defined by 35 domain experts. These include some hardware features specific to the Hyundai Co. Ltd industrial plant but most are quite generic so can be applied to other plant environments. As shown in Table 3.1, I identify three types of features depending on their scope: hardware-based feature, time-based feature, and size-based feature.

Table 3.1 Features can be grouped into three classes having as scope the hardware, time, and size

Scope	Feature	Description
Hardware-based feature	Alarm ID	• Alarm id represents the message type that was produced in the alarm data.
	Facility ID	• The facility id shows the identifier for each facility in the industrial plant.
Time-based feature	Time	• Time feature represents the starting time of the specific alarm.
	Lifetime	• The lifetime describes the length of time that the specific alarm is alive in one hour.
Size-based feature	Count	• Count feature shows the occurrence of the alarm data in one hour.
	Ratio	• Ratio feature represents the percentage of resources taken by the specific alarm.

Hardware-based features consider the individual hardware type in the industrial plant. It includes each alarm id and facility id. Alarm id represents the message type that was produced in the alarm data. The facility id shows the identifier for each facility in the industrial plant.

Time-based features consider the characteristics of time factor for each alarm data. It contains time and lifetime of alarm data. Time feature represents the starting time (e.g. 17 means 5pm) of the specific alarm. The lifetime describes the length of time that the specific alarm is alive in one hour. The length would be described as millisecond.

Size-based features consider the size of each alarm data. It includes occurrence and capacity of the specific alarm. Count feature shows the occurrence of the alarm data in one hour. Ratio feature represents the percentage of resources taken by the specific alarm.

Based on those three features, I produced 6 individual features (alarm_id, facility_id, time, lifetime, count, ratio) into the training dataset for a supervised classifier. Some example alarm training data are shown in Table 3.2.

In the table, I demonstrated 7 first alarm data in the training dataset. Each row represents 6 different conditions/attributes and its failure status of a specific alarm.

Table 3.2 The sample training data: first 10 rows

Alarm ID	Time	Facility ID	Count	Lifetime	Ratio	Status
DRV 183	17	H1103364	1	3228	896.67	INTRUDE
ES 041	16	H1101349	10	112	31.11	HUNTING
MCC 323	23	H1103364	1	3600	1000	IMPACT
APC 014	8	H1101349	4	22	6.11	BUR
PAG 004	1	H1101613	13	43	11.94	LEAK
PRC 090	9	H1101349	4	21	5.83	CARBONIZATION
PRC 058	7	H1105709	1	30	8.33	NORMAL
PRC 071	22	H1102579	1	82	22.78	NO LINK
GRS 008	10	H1105709	1	20	5.56	NO REVERSE
PRC 020	7	H1101613	1	4	1.11	CUT
...	...	...	...	...	...	...

3.3 Summary

In this study, the actual alarm data used in the Hyundai steel factory was analyzed and evaluated for conformity to the knowledge using machine learning.

An alarm collection interface was developed to collect 567,748 alarms during the year (September 2015 to July 2016) and was analyzed alarm patterns for one week, one month, and five months, respectively. The analysis showed that over 100 alarms occurred over an hour, which is an untreatable amount of monitoring personnel. In addition, it can be seen that the alarms do not have a particular pattern of occurrence, which is a factor that can be used to know that change management for knowledge implemented using alarms is essential.

To apply such data to machine learning, feature selection and labeling in data attributes are essential. First, Pearson's Correlation, LDA, ANOVA, and Chi-Square method were used for feature selection. As a result, three types (total 7) of features were determined. 35 domain experts helped with data labeling, resulting in 50 labels.

The processed data is used as testing data/training data for the knowledge learning method proposed in Chapter 4, and knowledge is constructed using alarms for failure detection. The proposed method is compared with existing machine learning methods to evaluate the performance.

4 Failure Knowledge Acquisition and Maintenance

4.1 Introduction

In the process and manufacturing industries, there have been many efforts to produce higher quality products, reduce product rejection rates, and meet increasingly stringent safety and environmental regulations. To meet the highest standards, most of modern industrial plants contain large number of facilities interacting with thousands of sensors and control, and those detected sensor data can be managed by Cyber Physical System(CPS) While these facilities can compensate for many types of disturbances, there are changes in the process which the controllers cannot handle adequately. These changes are called as faults or failures. A single failure in a facility can produce inconsistent outcomes, which can harm the core part of the industrial plant that may cause a critical industrial disaster. Therefore, it is crucial to find and apply the best solution for maintaining facilities and preventing industrial disasters [1]. Failure and fault diagnosis is a key application that improves efficiency and productivity.

The early-stage solution was the regular manual maintenance by human workers but this approach cannot be a perfect solution to prevent most industrial disasters [2]. Firstly, because regular maintenance is not effective for all facilities and secondly, because it is very expensive and time consuming.

The recent trend of industrial plant failure detection applications focuses on two main factors, alarms and human expertise. The CPS collects the status of different types of facilities from the sensors, which are attached, on each facility. For example, Figure 4.1 shows the partial architecture of CPS in Hyundai Steel plant. If there is any specific symptom detected by sensors the alarm will be ringed. The collected alarm data is sent to human expert in real time. The human experts have experience of several types of industrial disasters which gave them sufficient knowledge in diagnosing and treating failures. Applying facility sensor network, alarm data and

human expertise seems to be a good combination in handling failure but this approach also has two key issues.

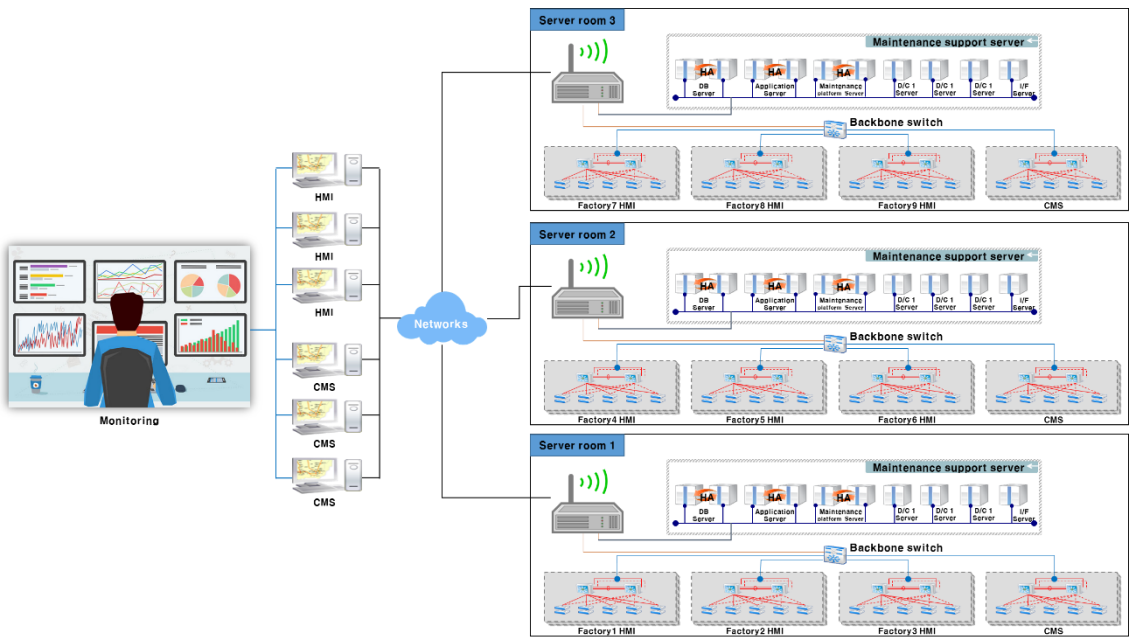


Figure 4.1 A partial architecture of Cyber Physical System in Hyundai Steel plant

Firstly, poor facility alarm and sensor network management may produce alarm flooding, which is the phenomenon of presenting more alarms in a given period of time than a human operator can effectively respond. The amount of the collected alarm is too enormous to be properly checked and handled by human experts. Owing to this, some severe failures can be misled or skipped, which may cause a critical industrial disaster. Alarm flooding has been identified as the root cause of significant plant incidents such, as Texaco Pembroke [126] and Three Mile Island Nuclear plant [127]. Several machine learning-based (rule-based or model-based) outlier detection algorithm [128, 129] was proposed in order to reduce human expert effort but it still requires further maintenance with algorithm and human experts.

Secondly, diagnostic and treatment activities are too depended on human experts. There are only limited numbers of human experts who have sufficient experiences in the certain industrial plant. There are two major issues, including availability and lopsided experiential knowledge. Human experts are not always available with every single situation. It cannot be always expected any proper treatment if human experts are not available. Additionally, human experts may have lopsided experiential knowledge. Different human experts can diagnose and treat a failure differently. Moreover, some failure cannot be diagnosed or treated since the expert have never experienced before [3].

In order to solve those issues, knowledge based systems are introduced [130] with data mining and human knowledge engineering. The aim of knowledge based system reasoning and using a knowledge base to solve some complex problems, such as prediction, detection, or recommendation. Most knowledge based systems are constructed by using two different approaches, machine learning technique and human expertise.

For the first solution, machine learning has been applied in order to manage knowledge for detecting failures. Machine learning techniques enable the system to acquire the knowledge from existing alarm data with no help of a domain expert. The techniques are very fast in finding the important pattern and knowledge from the provided data so it reduced the time and cost. However, machine learning has some drawbacks, such as over-generalization and over-fitting [4].

Another solution for failure detection knowledge based system was conducted with human experts. Human domain experts have enough experience so they can save knowledge in order to solve complex problems in a specific domain. However, knowledge acquisition from a human expert is normally in a slow pace. Even if the knowledge was acquired, the acquired expertise tends to be lopsided and would not cover the whole concept of knowledge in the domain since experts acquire domain knowledge based on their past experience [5].

To address this concern, the thesis proposes a new industrial plant failure detection approach that is able to leverage the benefits of machine learning and human expertise by using alarm data.

In order to achieve this, this research firstly collected various types of alarm data that detects a functional failure in Hyundai Steel factory over a one-year period (from September, 2015 to July, 2016). Based on this data, I recruit 35 domain experts in Hyundai Co. and ask them to select the feature and label the class for the training dataset. The training dataset acquires failure detection knowledge from machine learning and human experts by using Ripple-down Rules (RDR) based knowledge based system. The proposed approach generates knowledge through machine learning known as InductRDR and enables the maintenance of knowledge to be ascertained through human experts.

The contribution of this research can be summarised as follows:

- The thesis proposes an innovative approach to data-aided industrial failure diagnosis by using machine learning for the knowledge acquisition phase of a knowledge based system and human expertise for the knowledge maintenance phase.
- For failure detection in CPS applied large industrial plants, many studies have been conducted with using simple outlier detection, basic data-based machine-learning techniques, or human experts monitoring. The proposed approach produces the following benefits: (1) machine-learning generated knowledge base that removes the knowledge bottleneck and (2) the human expertise maintenance that enables for incremental learning and solves over-generalisation and over-fitting issue.

This research is organized as follows: In section "Failure Detection Framework", I describe the experiments of failure detection and proved the novelty of the proposed methodology. Finally, I conclude the research in section "Evaluation".

4.2 Ripple Down Rules (RDR)

In the knowledge engineering field, Ripple Down Rules(RDR) [131] is regarded as one of the best knowledge acquisition method for expert systems. RDR is able to reduce the knowledge acquisition bottleneck [132] and also enables resolving the verification process when domain users handle the validation themselves.

4.2.1 Single Classification Ripple Down Rules (SCRDR)

SCRDR stands for Single Classification RDR. The example of SCRDR knowledge tree can be found in the Figure 4.2. According to [131], the SCRDR structure is a finite binary tree where each node can have two distinct branches, which are called except and if not. Examples are measured from the root node of the SCRDR tree. Each node in the tree is a rule with the form of if **α** then **β** (**α** is the condition and **β** is the conclusion). If an example satisfies the condition **α**, it is passed to the next node of the except branch. Otherwise, the example is passed to the next node following the if not branch. If an example satisfies **α** but the node does not have the except branch, **β** of this node is the conclusion for the example. If an example does not satisfy **α** but the node does not have the if not branch, **β** of the last node on the path where the example satisfies its **α** is the conclusion for the example. In order to ensure that a conclusion is always returned, examples always satisfies the condition of the root node. This node is called the default node and the conclusion is called the default class. For instance, as in Figure 4.2, Node 1 is the default node and class '0' is the default class. An example which only satisfies condition A should be passed down through Node 1 and stops at Node 2. Since Node 2 does not have the if not branch, the example is classified as '0' by Node 1. If an example satisfies A, B, C and D, it should be passed down through Node 1, Node 2, Node 3 and stops at Node 4. Since it satisfies the condition of Node 4, it is classified as '1' [133].

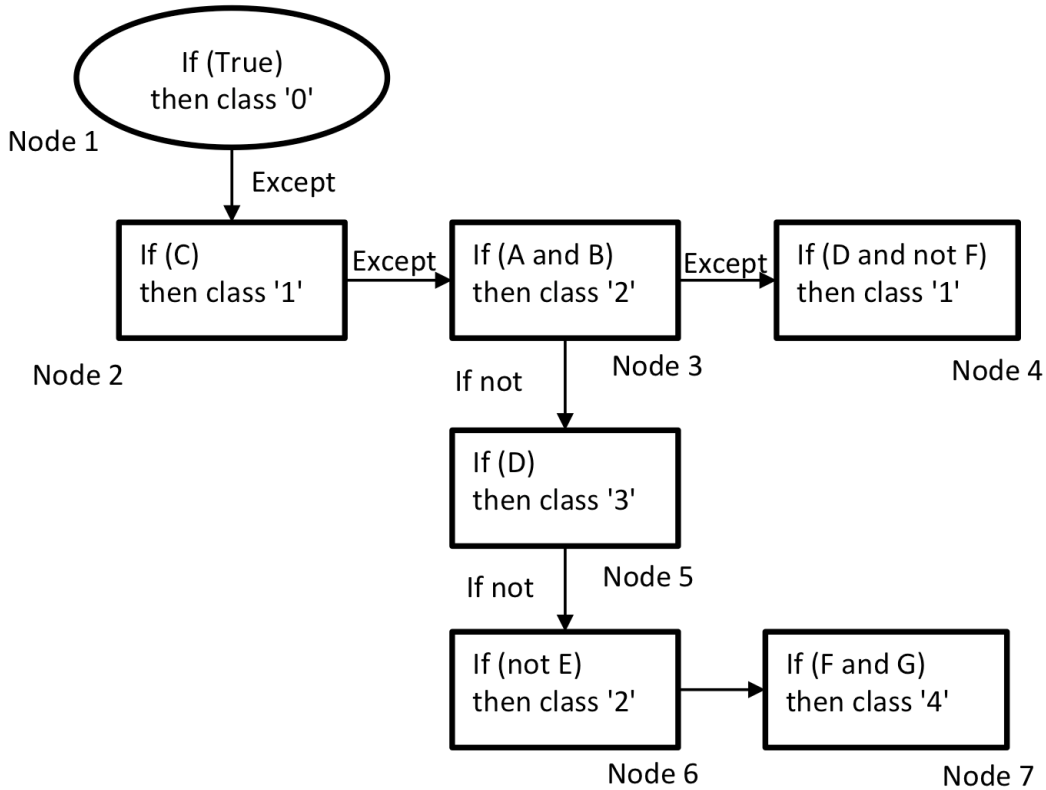


Figure 4.2 An example of SCRDR knowledge tree

When the measure process returns the wrong conclusion for an example, a new node is attached to the last node in the SCRDR tree. If the last node has no except branch, the new node is attached as the except branch, otherwise it is attached as the if not branch. The example which is associated with the new node is called the cornerstone example for that node. The rule generated for the new node entails the features of the cornerstone of the new node but not that of the cornerstone of the last node where the new node is attached. When the measure process returns the wrong conclusion for an example, a new node is attached to the last node in the SCRDR tree. If the last node has no except branch, the new node is attached as the EXCEPT branch, otherwise it is attached as the IF not branch. The example, which is associated with the new node, is called the cornerstone example for that node. The rule generated for the new node entails the features of the cornerstone of the new node but not that of the cornerstone of the last node where the new

node is attached [2].

In SCRDR, all rules are constructed in a binary tree. When the system encounters an incorrect classification, a new exception rule is added based on expert judgement. Therefore, SCRDR can incrementally develop a relatively accurate knowledge base, provided the domain is fixed and the experts provide the correct judgements.

Since RDR based knowledge base depends on expert judgement, the correctness of the used language expressed by the expert is the key of developing a good knowledge base. According to Pham and Hoffmann [131], it may cost a long time to classify most of the relevant cases correctly, if the target is linear threshold in the numerical input space and an expert is only allowed to use axis-parallel cuts, since it is unsuitable for him to express accurately.

4.2.2 Multiple Classification Ripple Down Rules (MCRDR)

Kang el al. [134] introduced Multiple Classification RDR (MCRDR) as an extension of RDR (SCRDR) to improve the limitations of RDR (SCRDR) including reducing the burden of the knowledge acquisition task and preventing knowledge base being ill structured which may result in considerable repetition of knowledge.

Unlike SCRDR, MCRDR evaluates all the rules in the first level of the knowledge base. The rules of the second level are evaluated to refine the rules which are satisfied at the first level. It keeps evaluating the next level in a recursive way until there is no more level to evaluate or none of the rules can be satisfied [135, 136]. MCRDR is able to provide multiple conclusions since it constructs rules with multiple paths. Each path is a particular refinement sequence. Knowledge is acquired from the experts when an example is classified incorrectly or needs to be given a new classification. The process can be described in the following three steps. 1) The expert provides correct classifications for the examples of the system, 2) The system decides on the location for the new rules, and 3) New rules are provided to the system by the expert and added to the

knowledge base for correction.

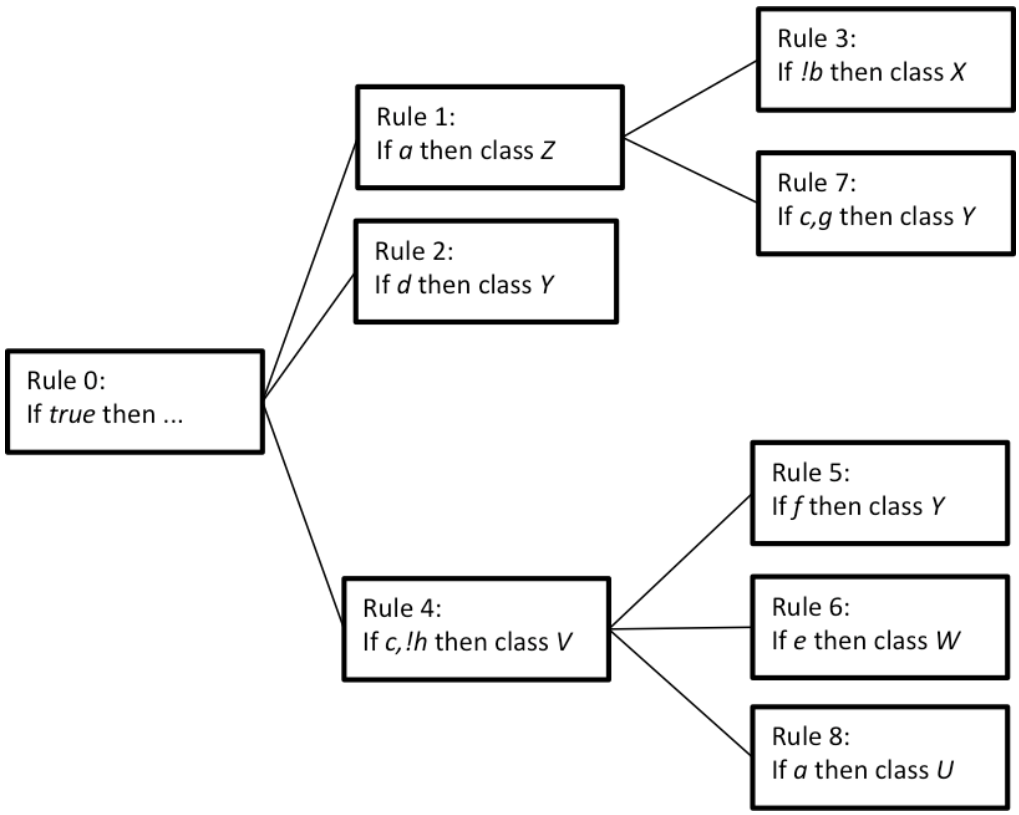


Figure 4.3 An example of MCRDR knowledge tree

The expert selects valid conditions from the current example to acquire the new rule for a given classification. The rule that has been created is then compared with the cornerstone cases of each node. If any cornerstone cases of a node satisfy this new rule the expert needs to select extra conditions for differentiating the current case and the cornerstone cases. For example, in Figure 4.3, when a case only satisfies conditions a and c but its correct class should be W, the system may decide the new rule location is on either Rule 3 or Rule 8. Since cornerstone cases of Rule 3 and Rule 8 are found to satisfy the conditions, experts are required to provide extra rules to cause those cornerstone cases no longer to satisfy the set of conditions. The system then repeats

this process until no remaining cornerstone cases satisfy the rule and it may simply add a new classification which is not in the tree.

There are three ways of correcting the knowledge base [134].

1. Add a stopping rule at the end of a path to prevent the wrong classification
2. Add a rule at the end of a path to give the new classification.
3. Add a rule at a higher level to give the new classification.

As can be seen in Figure 4.3, the system should add a new rule at the end of the path (Rule 3 or Rule 8) to give a new classification.

MCRDR concerns multiple independent classifications, whereas it maintains the advantages and principles of SCRDR. Like SCRDR, MCRDR is also based on the premise that a justification experts provide is necessary for a correction of knowledge in a particular context. However, the context in MCRDR is maintained in a different way and only consists of rules that have been satisfied by the data. Besides the validation of MCRDR includes differentiating the new example from a range of different examples.

A class in a MCRDR tree is the set of separated rule paths which provides the same conclusion. For example, in Figure 4.3, Class Y has three rule paths: Rule (0, 1, 7), Rule (0, 2) and Rule (0, 4, 5). Therefore, a rule path consists of all conditions of all previous rule nodes and conditions of the last node which concludes the class. Han et al. [132] mentions during the process of building MCRDR structures, the relationship between different classes are untouched and invisible from users. However, this information is able to provide inspiration for users to capture the point of rule creation and help users to realise how relationship may change the meaning or importance in the domain. Although MCRDR can work very effectively in many domains, similar implicit information contained within the structure itself is still not being extracted or exploited simply.

4.3 Knowledge Management by Machine Learning

As mentioned in the previous section, knowledge acquisition is traditionally conducted with human domain expert and knowledge engineers. However, there are two major issues in acquiring knowledge from domain experts: first, knowledge acquisition from human experts is normally in a slow pace; secondly, an expert cannot cover whole concept of knowledge in a specific domain. Because of those issues, it is almost impossible to manage the demand of expanding knowledge since a successful knowledge base may require an extremely large number of concepts and rules.

Machine learning techniques received lots of attention since those can learn and acquire concept and knowledge from the existing data with no domain experts' help in a short period time [137]. The most common machine learning techniques for knowledge discovery are neural network and decision tree.

Neural network models the human brain and consists of a number of artificial neurons and connections. Cascading chains of decision units with neurons used to recognize non-linear and complex functions. Knowledge can be acquired based on the input data incrementally so it does not need to be re-programmed. Only training phase is required in order to maintain the knowledge base. Time-series alarm data for failure detection was applied with neural network learning model so the model achieved a highly successful rate even though it has some noisy and outlier data. Tjhai et al. [138] focused on filtering alarms using the combination of neural network and k-means clustering.

Decision Tree algorithm is the typical machine learning approach that builds the knowledge base with the interpretable tree-structured rules. The Nearest Neighbor model is assigned to the most common class among the data samples that are most similar to the newly presented data. The approach has been used with small size of alarm data because of its extremely large distance calculation time. Both decision tree and nearest neighbor algorithm have been used a lot in the sensor failure detection and prediction since it is easy to understand the consequences, which can

trace the result. Chen et al. [139] designed and implemented a failure detection system using a decision tree learning approach, which is fast and easy to interpret.

However, those machine-learning techniques have over-generalization and over-fitting issues if the size or range of data is not sufficient to cover the knowledge in the specific domain.

In order to solve this issue, Gains [140] introduced Ripple Down Rule (RDR) based machine-learning technique, called InductRDR. The purpose of InductRDR is combining the concept of knowledge creation through machine-learning technique and knowledge acquisition from human domain experts. Gains described a sequence of dispersing knowledge partially from the view of a human expert, which consists of the following seven stages: Minimal Rules, Adequate Rules, Critical Cases, Source of Cases, Irrelevant Attributes, Incorrect Decisions, and Irrelevant Attributes & Incorrect Decisions [141].

The first stage is a complete, minimal set of correct decision rules so no data is required for knowledge acquisition since the correct answer is available from the expert. On the contrary, the last stage is a source of data from which the correct answer might be derived with the greatest probability of correct decisions so the expert has provided little. The stages in the middle from top to bottom show a decrease in existing knowledge though human intervention but an increase in new expertise through machine learning [140].

The main use of existing RDR is close to the top stage. Therefore, Induct RDR that derives rules directly from an extension of Cendrowska's Prism algorithm was made to be close to the bottom [142]. This Induct RDR sums standard binomial distribution as the possibility of selecting the correct data at random to measure the correctness of a rule.

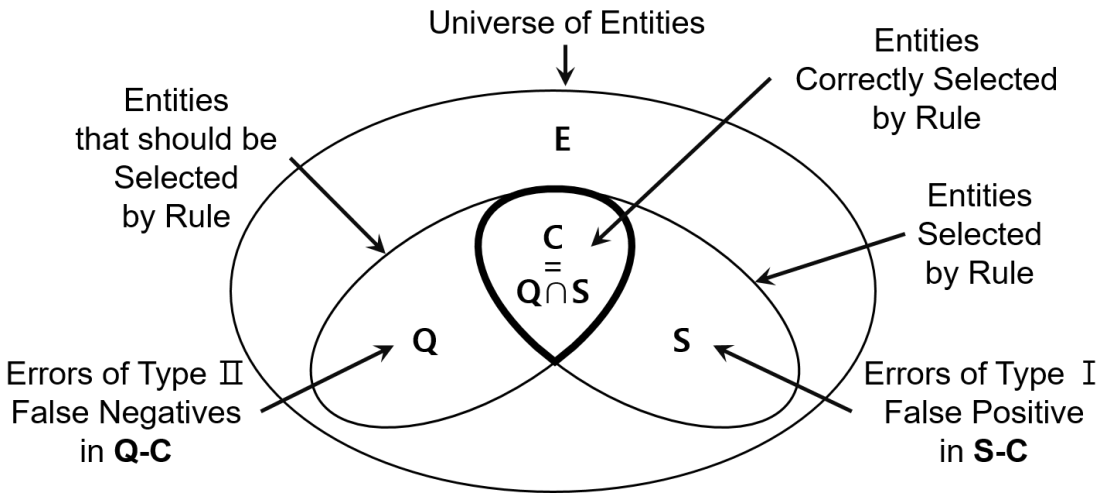


Figure 4.4 Problem of empirical induction

In Figure 4.4, given a universe of entities E, a target predicate Q and a set of possible test predicates of the form S on entities in E, use them to construct a set of rules from which the target predicate may be inferred given the values of the test predicates. The probability of selecting s and getting c or more correct at random is the sum of the standard binomial distribution.

In supervised learning, there is a risk of over-fitting the noise by memorizing the peculiarities of the training data [84]. Pruning approaches are commonly applied to solve the problem. Although Induct RDR recognizes the importance of pruning, it only removes redundant clauses and compresses the structure to some extent. Reducing over-fitting and improving generalization prediction capability has not been considered [143]. Ripple-Down Rules classifier (Ridor) is an implementation of Induct RDR in Weka. It first creates the default rule. The exceptions are created for the default rule with the lowest (weighted) error rate [144]. Different from the original Induct RDR, Ridor applies information gain to evaluate each rule and it prunes a rule by reducing error pruning.

4.4 Failure Detection Framework

The goal of this research is to propose new failure detection framework for industrial plant by using alarm data and RDR knowledge based system. The proposed RDR knowledge-based system for detecting failure allows acquiring knowledge by applying machine learning technique and maintaining them by domain experts who have experience in detecting failure from large industrial plants. The proposed framework can be described in the diagram.

This research would like to briefly introduce the proposed failure detection framework before describing the detailed process. The proposed framework can be seen in Figure 4.5.

First, I built a training dataset with 6 features/attributes and a 'status' class as described in the previous section. Then, using the training dataset, I built a supervised classifier by using RDR-based machine learning, Induct RDR. InductRDR adopts knowledge acquisition approach of the traditional machine learning techniques, which allows creating a knowledge base from the structured training dataset, but produces the rule-based knowledge base in Ripple Down Rule format. Therefore, InductRDR would enable human domain experts to modify the existing knowledge base, which is developed by machine learning technique. For example, if there is any incorrect classified data based on the testing dataset, human experts can add exception rules (either additional or refine rule) where data are incorrectly classified.

Then, it finds out incorrectly classified data based on the given testing dataset. The knowledge based system was acquired rules from human experts to add exception rules (additional rule) where data are incorrectly classified.

The following sections, "Knowledge Acquisition by RDR-based Machine Learning" and "Human Knowledge Acquisition using RDR Framework", include the detailed process of knowledge acquisition with InductRDR and knowledge maintenance with human experts. Note that I have updated several functionalities of original InductRDR [140, 145] in order to achieve better performance with the large size of real-time alarm data.

Training Dataset

Attribute						Class
Alarm ID	**Time**	**Facility ID**	**Count**	**Lifetime**	**Ratio**	**Status**
DRV_183	17	H1103364	1	3228	896.67	INTRUDE
ES_041	16	H1101349	10	112	31.11	HUNTING
MCC_323	23	H1103364	1	3600	1000	IMPACT
APC_014	8	H1101349	4	22	6.11	BUR
PAG_004	1	H1101613	13	43	11.94	LEAK
PRC_090	9	H1101349	4	21	5.83	CARBONIZATION
PRC_058	7	H1105709	1	30	8.33	NORMAL

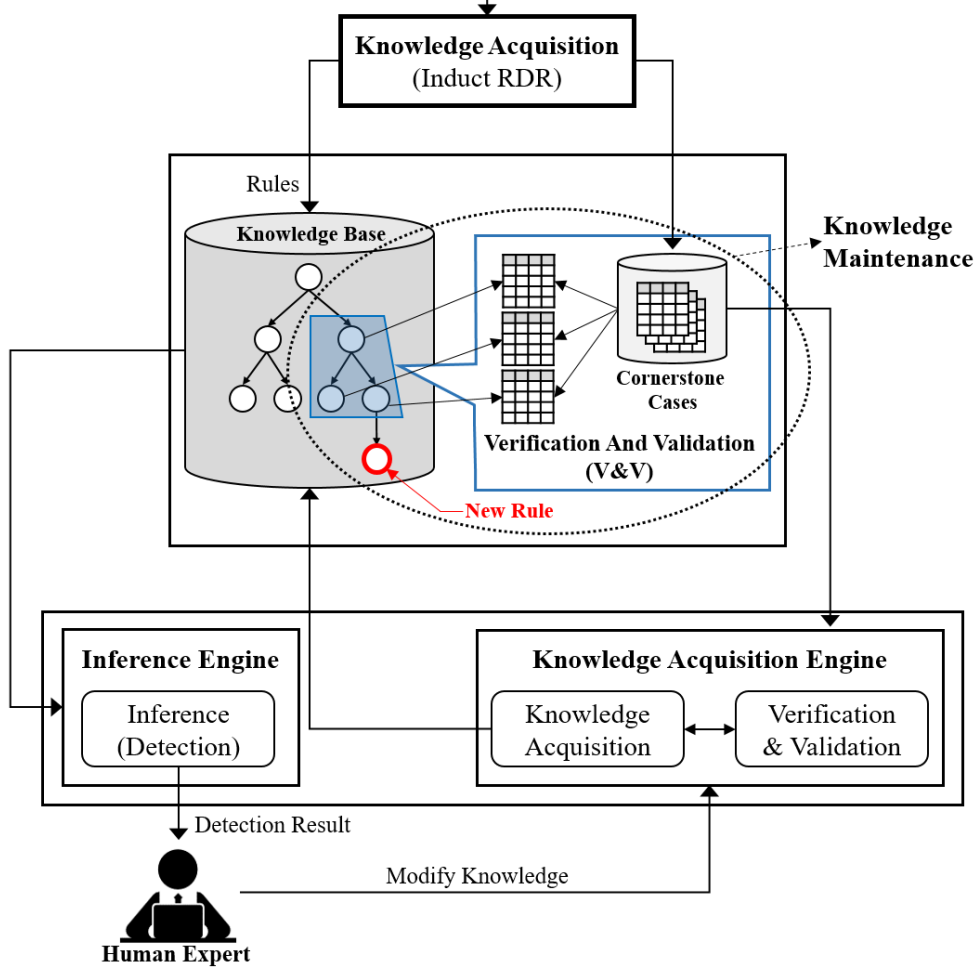


Figure 4.5 The proposed failure detection framework

4.5 Knowledge Acquisition by RDR-based Machine Learning

First, this research would like to discuss how it became possible to build the rule-based knowledge system with alarm training dataset by using the updated InductRDR. The basic idea of InductRDR is generating rules in a RDR structure with a rule induction algorithm. A rule at a single node in RDR structure is called as a clause, and it includes one or more terms in a form of attribute-relation-value.

Rule generation process of InductRDR can be described in the following three steps: First, the most frequently occurring class in the training data is selected as the default class value for the root-level rule. Then, it applies standard binomial distribution and searches a class that has the smallest m-value. The selected class is used for splitting the dataset into two subsets: true and false cases. If either of these two subsets has more than one class, the rule will be generated recursively. However, the original InductRDR does not fit into the alarm training data because of its size and complexity.

This thesis proposes three core updates in the original InductRDR. First, I updated the clause selection mechanism. The original InductRDR searches all possible combinations of terms in order to find the best class. In this process, m-value is an indicator that shows the quality of a term. Only appropriate terms are added to the clause until it only selects true positive data. Unfortunately, this process would produce severe computational issue if the domain has a large training dataset. In order to solve this issue, the updated InductRDR ordered terms first. Since m-value is used as a quality assessment function for each term and only terms with smallest m-value can be added to the clause, terms can be sorted by m-value in ascending order. The possible best terms will be always combined and assessed at the early stage, and that allows finding the best clause in a short period of time.

Algorithm 1: Procedure of calculating best clause

```
     Input : Class Value, Attribute, Training dataset
     Output : Best Clause
1    initialization
2    SET Term as default term
3    SET Clause as default clause
4    SET c as Number of examples of Class value in Training set for default class is true
5    SET t as Number of examples in Training set for default clause is true
6    repeat
7        repeat
8            Term ← term with attribute A and value V, that when added to Clause,
                    minimises m(Clause)
9            Clause ← Clause + Term
10           REMOVE A FROM Attributes
11           c ← Number of examples of Class value in Training set f or which clause is
                 true
12           t ← Number of examples in Training set f or which clause is true
13       until z = s
14       while m(Clause) > m(Clause - last term) do
15           Clause ← Clause − last term
16           RETURN Clause
17       end
18   until Find the best clause
```

Secondly, I modified the approach to evaluate the best clause. The original InductRDR applied m-function, the sum of the standard binomial distribution, for assessing the credibility of the clause [140]. Algorithm 1 represents the procedure of finding best clause by applying m-value. *Terms* are validated, and then the qualified terms are kept including into the clause, which is the combination of the terms. It would stop adding the terms when it only selects true positive examples [c=t]. Gains mentioned that m-value would produce the probability that the rule could be good at random, and that it derives no assumptions about sampling distributions. However, the problem would be occurred if the size of the dataset were too large. The m-values for all rules become to 0 with the big size of dataset so it is almost impossible for distinguishing the importance of the rules. This is because the original InductRDR just chose the attribute randomly

in this case. In order to remove this random selection, I borrowed the attribute selection approach, information gain, from decision tree learning algorithms since it is the key to improving prediction accuracy in decision tree algorithm [146]. The updated Induct RDR would use information gain for the best clause evaluation when m-value becomes 0.

Thirdly, this research adopts the numeric data handling approach in the updated InductRDR. While nominal data has fixed values with specific meaning, numerical data is usually continuous and the meaning is not clear. Nominal data can be divided into groups by their values but it is almost impossible to do the same thing for numeric data. InductRDR uses only inequality signs for best clause selection but it is extremely clumsy with large and complied dataset. Due to the nature of InductRDR, I applied information gain for numeric value handling.

Algorithm 2 : Chi-square Test Procedure

```
    Result : Chi-square Test result
    Input : Training Dataset
    Output : Chi-square Test result
1   n ← Length of training dataset
2   r ← Upper bound f or the random range
3   if n <= 10 * r then
4       Return False
5   end
6     /* PART A: Get frequency of data */
7   n_r ← n / r
8   h_t ← Frequencies of data
      /* PART B: Calculate chi-square */
9   sum ← 0
10  foreach ht do
11      sum ←sum + (ht − n_r)^2
    end
12  chi square ←sum / n_r
```

Finally, it is also crucial to reduce the over-fitting problem in the produced predictive modeling algorithm. Machine learning researchers usually applied pruning technique in order to

reduce the complexity of the learned model so that it is able to improve the predictive accuracy. The proposed algorithm applied chi-square technique in order to remove the rules that provide little power to classify the example. Algorithm 2 shows that the way to calculate the frequency of data (h_t) and calculate the simple chi-square by using frequency and length of the data. The chi-square originally proposed for reducing the over-fitting and over-generalization in the decision tree algorithm but the proposed rule-based hybrid algorithm is also perfectly matched with chi-square since the basic process would be similar.

With the above updates, I built the system that includes the updated InductRDR for knowledge acquisition and the interface for knowledge maintenance with human expert.

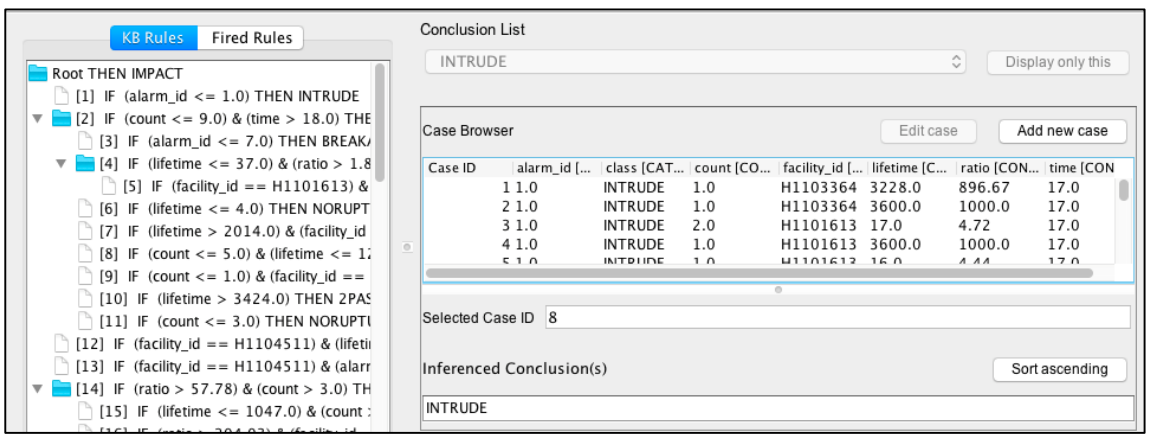


Figure 4.6 The generated knowledge base with InductRDR

I put the alarm data, which is collected and processed in the section "Data Collection", into the updated InductRDR, and it produces the RDR-structure rule (knowledge) base. This knowledge base can be seen on the left side of the Figure 4.6. It took 3 seconds to build the knowledge base, and has 177 rules in total. The prediction/classification performance of the updated InductRDR would be discussed in the section "Evaluation".

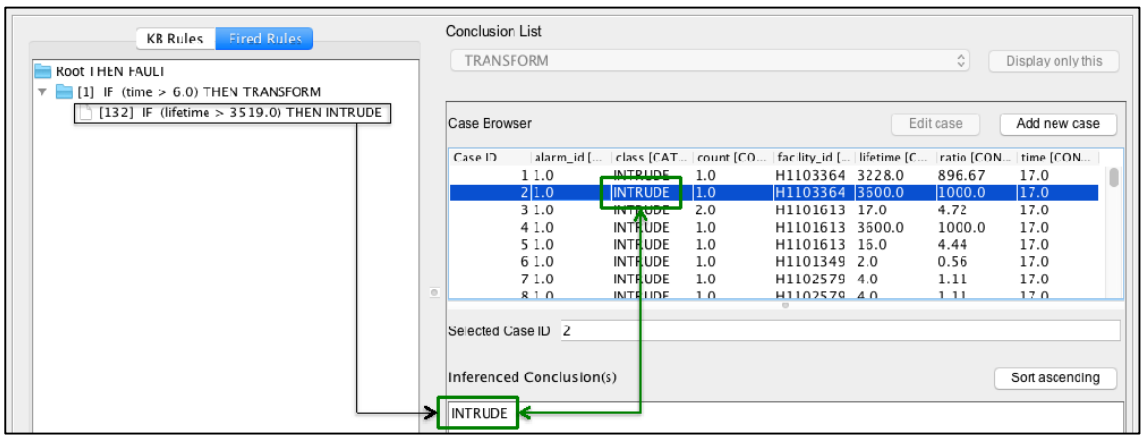


Figure 4.7 An example of Correctly Classified Instances

The right side represents the list of conclusion (class) list, and the case browser. The case browser shows the training dataset; each case is a row of the alarm dataset that I trained with InductRDR. The first row contains the value of 6 attributes (alarm_id, count, facility_id, lifetime, ratio, and time) and a value 'INTRUDE' as a class 'status'.

Figure 4.7 shows the correctly classified instance in the knowledge base. The value of a class for the second case (CaseID==2) in the case browser is 'INTRUDE' which is predefined. From the knowledge base, the rules contain the condition, which is matched with any value of 6 attributes, would be fired. As you can see the left size of Figure 4.7, the rule number 1 and 132 are fired for the selected case, case id 2. The rule no. 1 should be fired if the time is later than 6am, and the value of time attribute for the case is 17(5pm). The facility status should be classified as 'TRANSFORM'. However, before concluding this classification, the system checks the current case with the child rule no.132. The rule no.132 contains a condition to check whether the lifetime is over 3519 milliseconds, and the value of lifetime for the case is 3600 milliseconds, which has satisfied the condition. The final conclusion would be 'INTRUDE' as there is no more child rule to check. Therefore, the final inference conclusion and the original conclusion are equal, which means 'correctly classified'.

4.5.1 Human Knowledge Acquisition using RDR Framework

However, not all cases are correctly classified. As has been mentioned repeatedly in this thesis, one of the challenges in machine learning is the fact that not all instances will be classified correctly, a byproduct of issues such as over-fitting and over-generalization. Figure 4.8 shows an example, which is incorrectly classified. The proposed RDR framework system supports the function, which enables acquiring the human expert's knowledge based on the current context and adding that knowledge incrementally. As can be seen the Figure 4.8, the case id 121 produced the 'NOSTOP' status as a conclusion since the rule no.1, 128, and 131 were fired. The last child rule includes the condition to check whether the alarm occurred more than 3 times and the lifetime is equal or shorter than 168 milliseconds. The values of case id 121 are matched to the conditions but the class value 'IMPACT' does not match with the inference conclusion 'NOSTOP'.

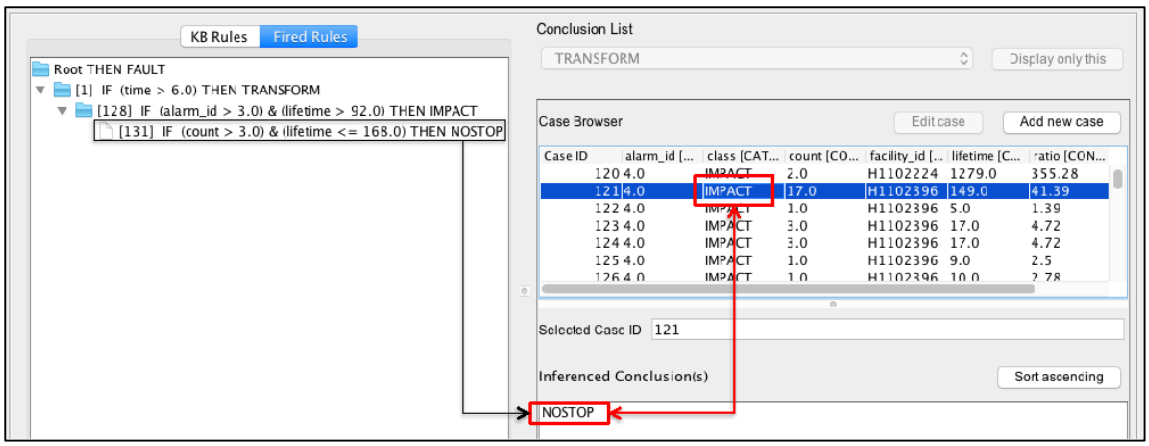


Figure 4.8 An example of Incorrectly Classified Instances

In this case, the RDR framework will acquire the rules from human experts for refining the knowledge base where the data is incorrectly classified by adding new rules. Figure 4.9 shows the output after the refine rule addition. The new rule no.178 is added so it is now correctly classified as 'IMPACT'.

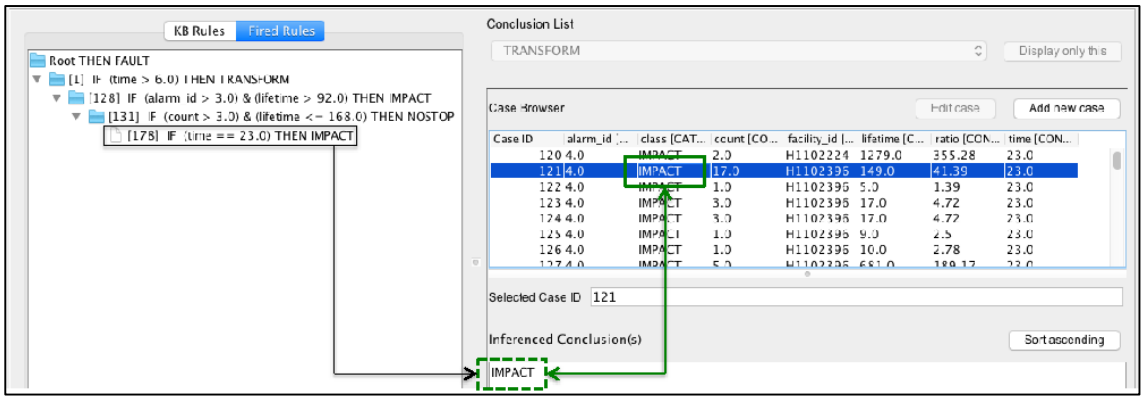


Figure 4.9 An example of Modified Rule

However, not all of the human knowledge can be applied. There are two reasons summarized as follows. First, there are data, which have the same vector of attributes but belong to different classes. This is because the existing attributes are not enough to tell the difference. Therefore, the class which the majority belong to will be decided at the conclusion and it is less possible to correctly classify the minority. Secondly, some rules applied might affect other correctly classified data. The knowledge created by the expert gives a hint about how these rules affect the whole dataset. If a rule has more incorrectly classified data than correctly classified data, it should not be applied.

The performance analysis of human rules addition will be conducted in the section "Evaluation".

4.6 Evaluation

In order to evaluate the performance of the proposed failure detection framework, I use 567,748 alarm data that was collected from a factory of Hyundai Steel Company, and processed by 35 human domain experts, employees in Hyundai Steel Co. The detailed collection and processing procedures are described in the section "Data Collection".

4.6.1 Failure Detection Performance Evaluation

To conduct the evaluation of our proposed failure detection framework, I compared the performance of the modified InductRDR on the alarm dataset against four other common machine-learning classifiers, including Naive Bayes, neural network, decision tree and support vector machine. The algorithmic approach and underlying philosophy of each of these algorithms are fundamentally different, however, each has shown to produce favorable results across different.

I tested the performance with six other machine learning techniques by using a 10-fold cross validation. The Table 4.1 describes the algorithms that are applied for the evaluations.

Table 4.1 Applied Machine Learning Techniques

No	Evaluation Technique	Base Algorithm
1	NaiveBayesSimple	Naive Bayes (NB)
2	MultilayerPerceptron	Neural Network (NN)
3	LIBSVM	Support Vector Machine (SVM)
4	C4.5 Decision Tree	Decision Tree (DT)
5	The modified InductRDR	InductRDR
6	The modified InductRDR + Human RDR rule	InductRDR and human expertise

The performance of failure detection with machine learning techniques can be found in the following table 4.2. In this domain, it shows that Neural Network and InductRDR achieved over 92% detection accuracy. In the case of RDR (machine learning and human rules), the knowledge base is built by Induct RDR before adding human knowledge. Then, the test dataset is used to examine this knowledge base to find incorrectly classified data. A simulated expert is used to find correct rules for those incorrectly classified data. In the case of InductRDR (machine learning only) and C4.5 Decision Tree, they are based on machine learning only so their prediction accuracy is based on predicting the test dataset using the knowledge base acquired from the training dataset. As Table 4.2 shows, the supervised classifier achieves an accuracy of 92%. The Kappa statistic indicates that the predictability of inductRDR classifier is better than a random predictor.

Table 4.2 The accuracy of failure detection with machine learning techniques

Evaluation Technique	**NB**	**NN**	**SVM**	**DT**	**InductRDR**
Correctly Classified Instance	80.02%	**92.31**%	87.53%	85.03%	**92.05**%
Kappa Statistic	0.58	0.52	0.54	0.51	0.88
Mean absolute error	0.19	0.14	0.14	0.16	0.08
Root mean squared error	0.38	0.30	0.28	0.29	0.26
Relative absolute error	48.31%	50.26%	51.55%	58.77%	21.47%
Root relative squared error	85.87%	82.14%	74.76%	103.86%	46.65%

As can be seen in Table 4.3, it has been found that the updated InductRDR only can achieve 92.05% of prediction accuracy. After adding human rules, the result can be improved up to 100%. However, upon updating the classifier with domain expertise, the prediction accuracy markedly improved, classifying all training instances with 100% accuracy. Therefore, one can surmise that

adding human knowledge to the knowledge base generated by the machine learning classifier does improve the classification accuracy and can mitigate some of the pressing concerns of machine learning as I can handle issues of noise or anomalous data to some extent. It is important to note that the 100% accuracy achieved with the incorporation of the human rules applies to the current fixed dataset and I would expect the accuracy to be reduced in a real-world clinical setting. However, the benefit of this approach is the ability to adapt through incremental learning and so the system is able to improve in the real-world settings, even when the performance reduces.

Table 4.3 The performance comparison with machine learning techniques and proposed InductRDR with human rules

Evaluation Techniques	**Detection Accuracy**
Neural Network	92.31%
The updated Induct RDR	92.05%
The updated Induct RDR with human rules	**100%**

Although Neural Network had the best prediction accuracy (92.31%) among machine learning techniques, the updated InductRDR with human rules outperforms it eventually. Therefore, it can be concluded that adding human knowledge to the knowledge base created by machine learning does improve the prediction accuracy. The prediction accuracy becomes low if there are significantly over-generalization and over-fitting problems. In this case, prediction accuracy has been improved so that it implied that over-generalization and over-fitting problems have been solved to some extent.

In addition to the high performance of failure detection, the proposed approach allows human experts to incrementally add and maintain the knowledge in the knowledge base with no rebuilding or initialization process.

4.6.2 Failure Detection Performance at Feature Level

In this section, I study how specific subsets of features perform in the task of failure detection. To do this, I train the InductRDR (machine-learning) algorithms considering subsets of features. I consider 3 subsets of features grouped as follows: 1) Time-based, 2) Size-based, and 3) Hardware-based. The detailed information on those feature levels can be found from the section 3.3.

I train the updated InductRDR with each subset feature at a training set. The instances in each group were split using a 10-fold cross validation strategy. In this evaluation, I aggregate all 47 faulty related classes as a "FAULTY" class while labelling 'normal' class as just "NORMAL" class as can be seen in Table 4.4. The results indicate that among the features, the time-based features and size-based features are very relevant to diagnosing the failure/faulty status. I observe that hardware-based features are not enough by themselves for this task. On the other hand, "NORMAL" class is in general more difficult to detect.

Table 4.4 Experimental results obtained for the classification of failure detection

Time- based					
Class	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F1**
NORMAL	0.623	0.082	0.610	0.623	0.616
FAULTY	0.918	0.377	0.923	0.918	0.921
W.Avg	0.868	0.327	0.869	0.868	0.869
Size- based					
Class	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F1**
NORMAL	0.147	0.033	0.478	0.147	0.225
FAULTY	0.967	0.853	0.847	0.967	0.903
W.Avg	0.828	0.714	0.785	0.828	0.788
Hardware- based					
Class	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F1**
NORMAL	0.749	0.454	0.643	0.749	0.692
FAULTY	0.546	0.251	0.666	0.546	0.600
Avg	0.652	0.357	0.654	0.652	0.648

I also applied ROC curve for comparing performance of diagnosing the failure/faulty status with size-based, time-based, hardware-feature-group and all feature groups based on the updated InductRDR. Figure 4.10 is shown to outline the accuracy of the prediction made by the updated InductRDR which was used in the training dataset. The x-axis is the false positive rate while the y-axis is the true positive rate. Closer the ROC curve gets to top-left part of the chart, better the classifier is. In Figure 4.10, it indicates that using all features have the highest accuracy rate. If all features are used, it takes multiple aspects into consideration at the same time, which gives a more uniformed result.

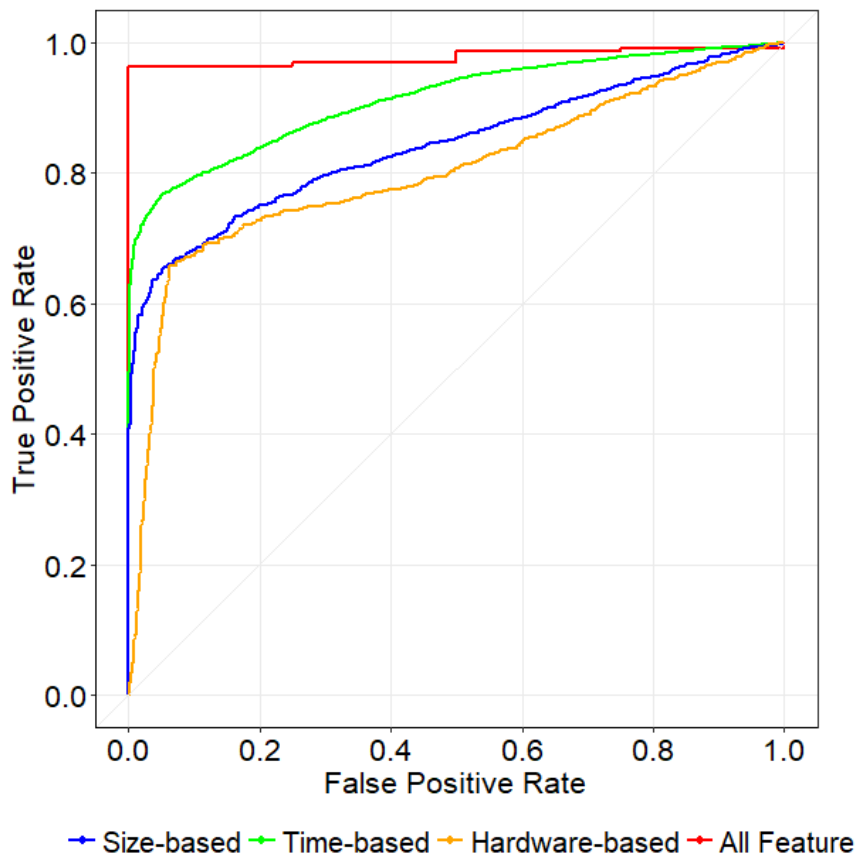


Figure 4.10 ROC Curve different subset using InductRDR

4.6.3 Cost Evaluation

In order to solve the core issue of machine learning, over-generalization and over-fitting is traditionally accompanied with inserting new data to the existing dataset to enrich the patterns. In this case, the previously created knowledge base will be removed, and a new knowledge base is constructed. The amount of knowledge can be quantified as the number of nodes and conditions in a knowledge base, so the cost of solving the problems can be quantified as how many nodes and conditions are reconstructed. This is the case of machine learning. In the case of adding human knowledge, the cost is how many nodes and conditions are added to the original knowledge base.

The following Table 4.5 summarises the result of reconstructed or increased nodes and conditions after solving over-generalization and over-fitting problems. By applying human knowledge, the increased ratio of nodes for improving 1% of accuracy is 28.57%, much smaller than InductRDR only (109.78%). Similarly, the increased ratio of conditions for improving 1% of accuracy is 60.15%, much smaller than InductRDR only (99.66%). As mentioned above, the reason that pure machine learning models cost much is because they remove previous knowledge base and create a new one every single time that it encounters a new data case which cannot be explained by the existing knowledge base.

Table 4.5 Cost Evaluation Result of Knowledge Increased

Models	**Updated InductRDR**	**Updated InductRDR with human rules**
Increased ratio of nodes	261.54%	58.25%
Increased ratio of conditions	222.58%	124.20%
Increased ratio of nodes per 1% of accuracy improvement	109.78%	28.57%
Increased ratio of conditions per 1% of accuracy improvement	99.66%	60.15%

Therefore, it can be concluded that the reconstructed or increased ratio of the knowledge base is much smaller by combining human knowledge and machine learning than those approaches based on machine learning only.

4.7 Discussion

Detecting failure status in large industrial plants has been noted as complex and dynamic problem area because of its enormous size of alarms and sensor data, and experiential knowledge requirements. Either machine learning technique or human expert system has been applied to acquire and maintain the knowledge for failure detection but neither did work successfully. In this project, I collected and analyzed the alarm data with 35 domain experts in Hyundai Steel Co., and propose a novel approach that uses Ripple-down Rule (RDR) to maintain the knowledge from human experts with knowledge base generated by the updated Induct RDR.

Based on the experiment, I found that it improves accuracy to 100% with the fixed dataset. It is important to note that the 100% accuracy achieved with the incorporation of the human rules applies to the current fixed dataset and I would expect the accuracy to be reduced in a real-world clinical setting. However, the benefit of this approach is the ability to adapt through incremental learning and so the system is able to improve in the real-world settings, even when the performance reduces.

4.8 Conclusion

The proposed approach in this thesis allows human experts to incrementally add and maintain the knowledge in the knowledge base without having to rebuild or re-initialise the knowledge base, unlike pure machine learning approaches which rebuild the knowledge base from scratch each time. Moreover, the proposed failure detection framework can reduce the time

of human expertise acquisition and the cost of solving over-generalization and over-fitting problems in machine learning technique. The proposed failure detection framework has never been reported previously. Moreover, this framework can be successful detection approach in the domain if it requires handling big size of the dataset and human expertise. Through the combination of: (1) machine learning to generate knowledge base that alleviates the knowledge acquisition bottleneck, (2) the human expertise maintenance that enables for incremental learning and (3) the mitigation of the failure detection problems reflected in previous research; I have confidence in this adoption of this framework across multiple modalities.

5 Process Map with Causal Knowledge

5.1 Introduction

This chapter introduces a natural language processing-based process map that can extract and manage knowledge that has a causal relationship by failure reports. In the industrial field, knowledge resources are documents written with expert knowledge.

First, expert knowledge is an acquired knowledge learned by experts. The research to acquire and manage this knowledge is introduced in detail in the previous chapter.

Second, the knowledge resources that occur in the industrial field keep recording form. In particular, the failure report is recorded by the specialist in terms of the failure situation such as failure, cause-effect, and corrective measures. This is an important reference for past failure cases as the cases can actually be utilized for fault analysis and measures in instances of a failure occurring first.

However, without a formalized form, it is the form of direct technical experts. In other words, the type that is described by experts as being quite varied with the knowledge in a form that a computer cannot understand. As humans should retrieve and analyze the stored failure reports directly, there is a higher reliance on human resources for fault management, so the handling crisis for the failure can be delayed, the industry that require rapid troubleshooting critical issues.

Therefore, there has been a demand for building a decision support system based on expert knowledge that is capable of automatically diagnosing and predicting failures in an industrial field.

However, this effort has not been successful because of the various changing factory environments, they could not reflect the changing knowledge and they could not build up the system for knowledge resources.

The main features are as follows:

First, using natural language processing techniques, the system automatically analyzes failure reports and translates them into structural knowledge with causal relationships. That is, the system is constructed in a usable knowledge form.

In addition, analyzed knowledge automatically builds a relationship with one another, based on existing knowledge and the degree of similarity. By doing so, it evolves to the knowledge of the network-type. It is possible to refer to similar and useful knowledge when handling current problems.

At last, analyzed knowledge provides UI that can integrate and edit both knowledge if there are similar cases in comparison with existing practices. This allows experts to easily modify and supplement knowledge in order to evolve. (Figure 5.1 is a schematic representation of the overall process.)

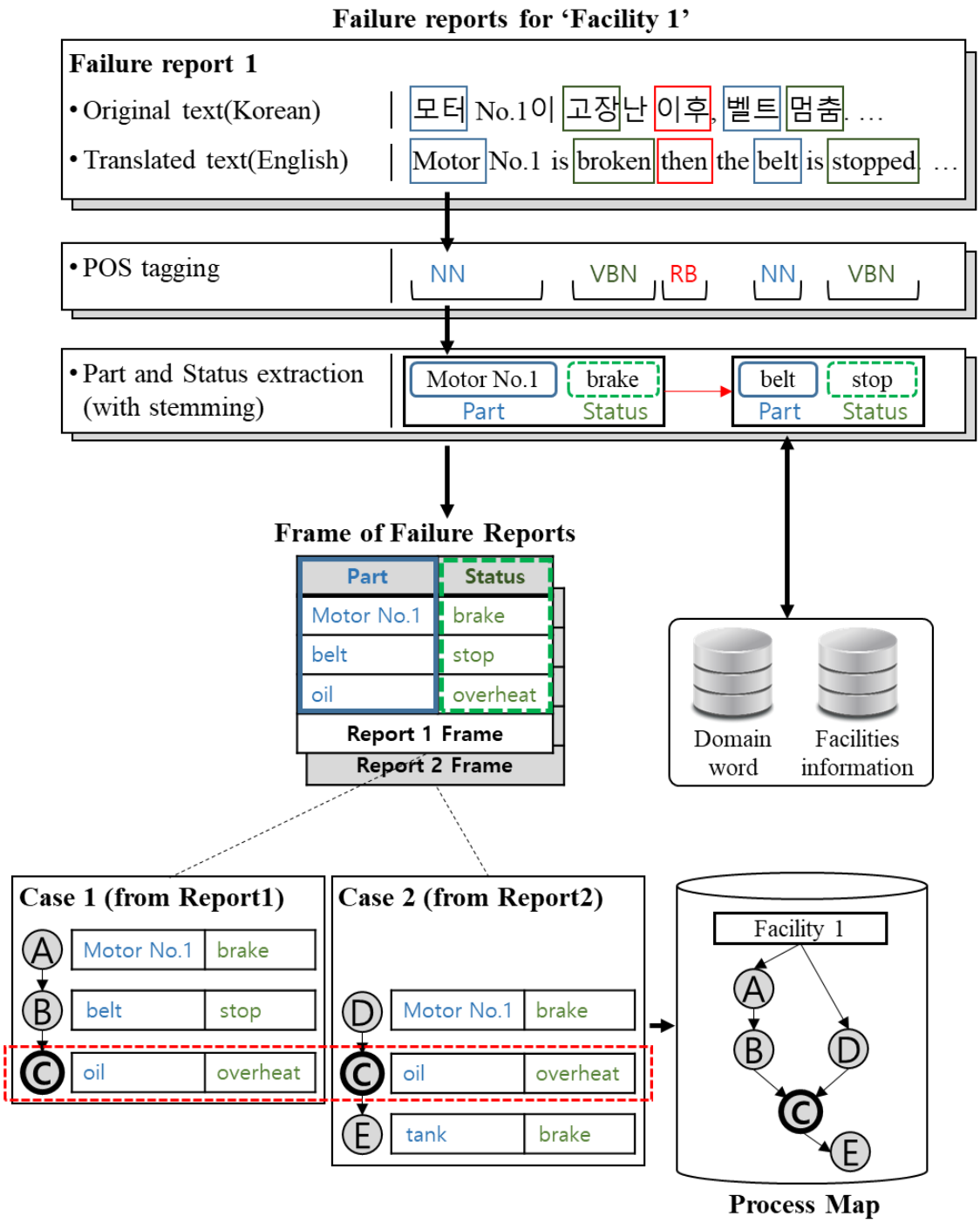


Figure 5.1 The proposed framework of Failure Report Analysis

Failure reports are analyzed and processed based on natural language processing techniques. The proposed method is based on the Korean Failure reports. Korean grammar processing algorithm and domain terminology dictionary are used for analysis. The sentence is separated into the shortest sentence, which is the minimum meaning unit, consisting of 'Part' and 'Status.' The relationship between short texts identifies the context, cause, and effect. It is connected to the link through postposition and conjunctions to connect the short texts. Through this process, a knowledge that has a causal relationship is completed. The failure cases are stored and managed in a knowledge structure called a process map. The process map is a graphical form of knowledge representation. Since the failure report is written about a specific facility, the process map also has each failure case based on the facility. When the analysis result is stored in the process map, if there is a similar relationship found by testing the similarity and the existing knowledge, it automatically builds a connection relationship with each other. Through this process, the process map evolves into graphical knowledge. In addition, the process map is easier to modify and extend knowledge than the ontology-based knowledge representation method, and a user interface is proposed to support it. Case 1 and Case 2, as shown in the Figure 5.1, may have the same knowledge that oil / overheat; the two cases are stored in it are said to bear a relationship to the process map.

Detailed methods and user interfaces for building process maps are described in the following sections.

5.2 Process Map Concept

The failure analysis system analyzes the failure reports in order to reuse expert's experiential knowledge. Failure reports include information related to the problem such as current status, cause, and actions taken, so that it can represent a causal relationship of the problem. Thus, it can be used as a knowledge base for failure diagnosis and prediction. Failure reports include domain dependent terminologies, implicit representation, and are written in a unique way for each writer. This is the reason why the system utilizes not only the failure reports, but also the domain terminologies and domain knowledge.

Knowledge is obtained by analyzing failure reports, extracting minimum semantic units from failure reports written in natural language, constructing causal relationships between them, and mapping the failure of the target facility. This knowledge is then labelled as failure knowledge. After that, failure knowledge is stored and managed in a process map. In the failure report, the simplest and basic type of sentence is the minimum semantic unit. Two components of the facility corresponding to the subject are extracted from the sentence: the subject that refers to the target of the facility and the predicate that means the status/operation of facility. The acquired unit knowledge with the form of node is called `failure phenomenon'. After extraction process, subject indicate as part of the component and predicate indicates as the status of the component. I analyzed failure reports and extracted short sentences from unstructured natural language. Order between two short sentences and consistency of meaning is considered while constructing a relationship which is referred to as failure case. A series of processes including how the specific facility has caused the problem and how the problem is solved, is configured in the order of occurrence through the relation between the failure phenomena.

Figure 5.2 shows the concept of a process map. The process map is based on the facility. There are numerous facilities in the factory, and many facilities have similar functions, but the type of facilities usually varies. Therefore, managing individual knowledge of these facilities can make knowledge generation very complex and knowledge management very inefficient. The proposed process map designates representative facilities with functions and designates the

facilities having similar functions after the representative facilities. Having a similar function means that the type and basis of facilities is similar, and the usage of facilities and the relation with other facilities are also similar. This knowledge model is suitable for complex domains such as large factories with various facilities. The failure report describes the results of failure analysis and treatment actions for the facilities that have a failure so that the failure cases analyzed from failure reports can be linked to these facilities.

In Figure 5.2, A and B are facilities with similar functions where each facility has one or more failure cases and is represented by a single pass. If failure cases have the same failure phenomenon, failure phenomenon is expressed as multipath constructed by integrating and sharing failure phenomenon.

For the similarity calculation, I applied following similarity formulation and text-based similarity algorithm:

A graph of the process map : $G_p = \{N_p, E_p\}$

- $N_p = \{n_{pi}\}$: Set of nodes in the process map
- $E_p = \{e_{pi}\}$: Set of edges in the process map

A graph of the failure report : $G_d = \{N_d, E_d\}$

- $N_d = \{n_{dj}\}$: Set of nodes in the failure report
- $E_p = \{e_{dj}\}$: Set of edges in the failure report

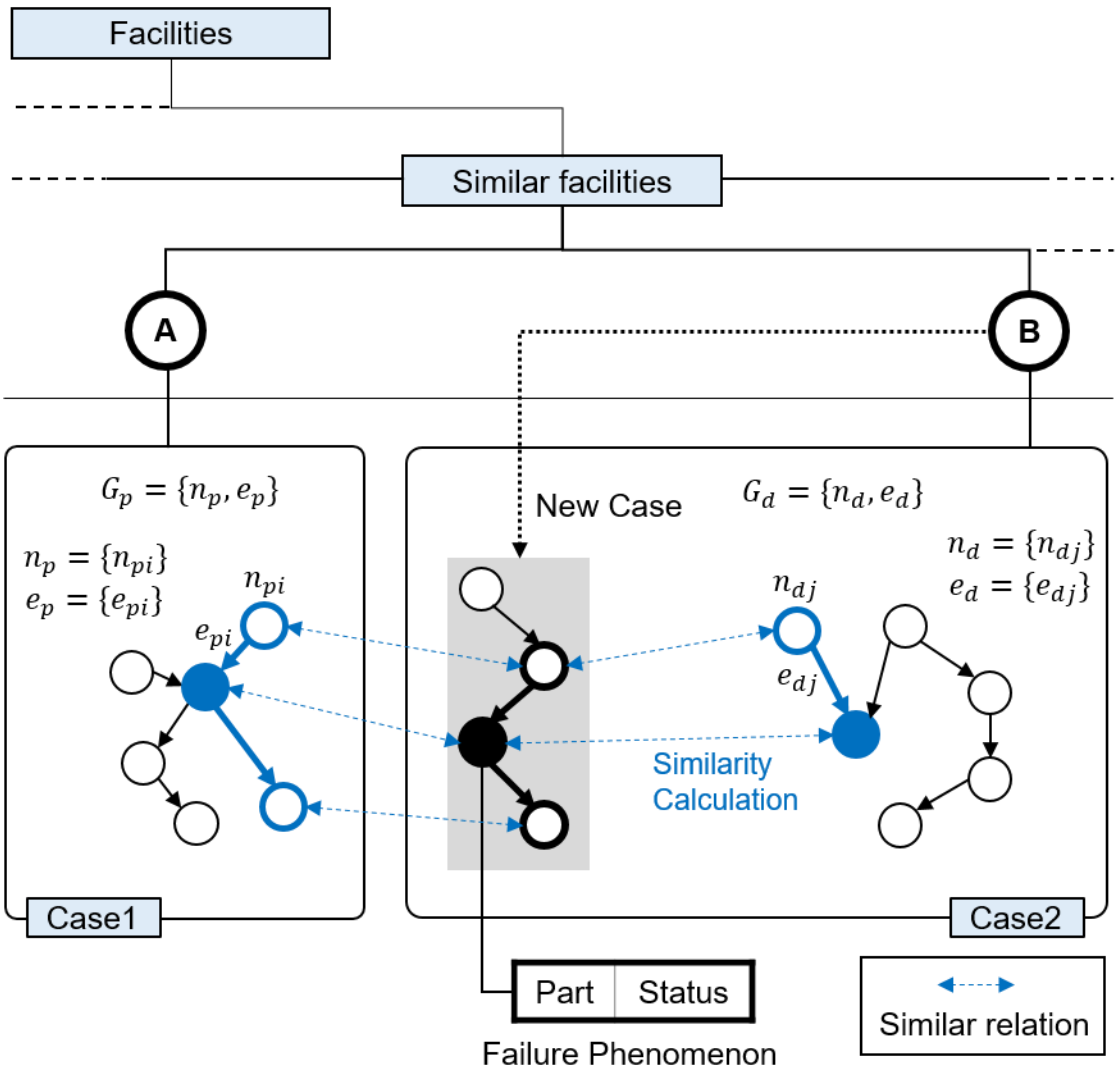


Figure 5.2 A conceptual diagram of the proposed process map framework

The process map construction principle consists of finding the node pairs that represent the same meaning among the nodes of the process map and the nodes of the failure report, which are represented graphically.

Finding a synonymous relationship:

- A pair of two short paragraphs with a value greater than a certain degree of similarity

$$Syn(n_{dj}) = \{\forall n_{pi} \in N_p | Sim(n_{dj}, n_{pi}) \geq threshold\}$$

$$Sim(n_{dj}, n_{pi}) = Sim_{Text}(n_{dj}, n_{pi}) + Sim_{Rel}(n_{dj}, n_{pi})$$

- Sum of the similarity of sentences that represent part/status included in each sentence and the similarity of cause/effect/action sentences connected to each sentence.
- Sim_{Text} : Similarity based on the strings of the nodes (representation of object / phenomenon)
- Sim_{Rel} : Similarity based on the relation of each node (reflects similarity of parent / child nodes)

Node similarity based on text:

- Reflect text similarity based on Edit-Distance

$$Sim_r(n_{dj}, n_{pi}) = EditDistance(n_{djr}, n_{pir}) + EditDistance(n_{djE}, n_{piE})$$

Node similarity based on relation:

$$Sim_{Rel}(n_{dj}, n_{pi}) = \max_{n_{qk} \in P(n_{qi}), n_{dl} \in P(n_{dj})} Sim(n_{pk}, n_{dl}) + \max_{n_{qk} \in C(n_{qi}), n_{dl} \in C(n_{dj})} Sim(n_{pk}, n_{dl})$$

- *P*(-) : a set of parent nodes, *C*(-) : a set of child nodes

Measure the likelihood of failure cases and scenarios:

- The similarity of all the failure phenomena belonging to the failure case and the similarity of all the failure phenomena belonging to the scenario (failure knowledge in the process map) are measured based on the Edit-Distance (the similarity of the phenomena is measured in the same manner as defined before).
- The similarity between the failure case and the scenario is defined as the sum of the largest similarity values of each failure phenomenon belonging to the failure case.

$$Sim(L_r, L_s) = \frac{1}{|L_r|} \sum_{f_i \in L_r} \max_{f_j \in L_s} sim(f_i, f_j)$$

L_r : *Failure case*
L_s : *Failure scenario*
f_j : *Failure phenomenon in failure scenario*
f_i : *Failure phenomenon in failure case*

Because similar failure phenomena of similar facilities have a relationship, whole structure of the process map is in a network. So, similar failure cases can be referenced in addition to direct failure case. In case that contents of failure knowledge are weak or non-existent, the process map that uses similar facilities and failure cases can easily be referenced in the indirect method. Thus, I can achieve high knowledge usability.

The operation to obtain the similarity between the case and the failure report on the process map is expressed in the algorithm. If the operation is briefly described, first of all, the similarity between the current node and the word is obtained. Next, to compare the similarity between cases, calculation is conducted by the similarity between the parent node and the lower node. That is, the similarity between the parent node and the child node is obtained, and the sum of the parent node and child node is compared to obtain the similarity between the case and the failure report.

Algorithm 3 : Edit Similarity Distance between sentences

Input: Two sentences
Output: Similarity distance between two sentence

LET Integer *DepthForMeasuring* as initial similarity
LET Integer *sim* as default similarity
LET String *str1* as the first string
LET String *str2* as the second string
LET Integer *str1Length* as length of str1
LET Integer *strLength* as length of str2
LET Integer *similarityOf Strings* as default similarity between strings
LET Integer *nodeSimilarity* as similarity distance of node1 and node2
LET Object *firstNode* as the first unit sentence node in a list
LET Object *secondNode* as the second unit sentence node in a list
LET Object *AncNode*1 as ancestor node for the firstNode
LET Object *AncNode*2 as ancestor node for the secondNode
LET Object *AncestorNodeSim* as node similarity between AncNode1 and AncNode2
LET Object *desNode1* as descendant of the firstNode
LET Object *desNode2* as descendant of the secondNode
LET Integer *DescendantNodeSim* as similarity between desNode1 and desNode2

DepthForMeasuring ← 1
sim ← 0
str1 ← *first string*
str2 ← *second string*
str1Length ←*length of str*1
strLength ←*length of str*2
similarityOf Strings ← 0

/* PART A: Get similarity distance between two string */
if *str1Length* ≤ *0 OR str2Length* ≤ *0* **then**
Return similarityOfStrings ← 0.0
else if *str1Length = str2Length* **then**
Return similarityOfStrings ← 1.0
else if *str1 is synonym for str2* **then**
Return similarityOfStrings ← 1.0
else if *str1 contains str2 OR str2 contains str1* **then**
return similarityOfStrings ←1.0
end
distanceOf String ← *distanceStrings(str1,str2)*
maxLength ← *Max(str1Length, str2Length)*
similarityOf Strings ← (1.0 − *distanceOf String*/*maxLength*)2
Return SimilarityOfString
/* PART B: Get max node similarity in list */

```
41  nodeLst1 ← getNode(The first list of unit sentence node)
42  nodeLst2 ←getNode(The second list of unit sentence node)
43  maxSimilarity ← 0
44  similarityOf node ← 0
45
46     /* Get node similarity */
47  foreach node1 in nodeLst1 do
48        foreach node2 in nodeLst2 do
49              Sim ← go to 8 to get similarity distance of node1 and node2
50              if maxSimilarity ≤ Sim then
51                    maxSimilarity ←Sim
52        end
53  end
54  Return maxSimilarity
55
56     /* PART C: Get similarity between nodes in list */
57  nodeSimilarity ← go to 27 to get similarity distance of node1 and node2
58  if nodeSimilarity ≥ 1 then
59        return 0
60  else
61  end
62  nodeSimilarity ← nodeSimilarity ∗ 2 / 3
63  Return nodeSimilarity
64
65     /* PART D: Get Ancestor Nodes Similarity */
66  firstNode ← getNode(The first unit sentence node in a list)
67  secondNode ←getNode(The second unit sentence node in a list)
68  AncNode1 ← getNode(Ancestor node for the firstNode)
69  AncNode2 ← getNode(Ancestor node for the secondNode)
70  AncestorNodeSim ← go to 41 to get max node similarity between AncNode1 and
71                          AncNode2
72
73    /* PART E: Get Descendant Nodes Similarity */
74  desNode1 ← getNode(descendant of the firstNode)
75  desNode2 ← getNode(descendant of the secondNode)
76  DescendantNodeSim ← go to 41 to get max node similarity between desNode1 and
77                            desNode2
```

Algorithm 3 calculates the edit similarity distance between sentences. It takes two sentences as input and then outputs their similarity distance. A variable DepthForMeasuring is set as 1 in the beginning and a variable sim to measure similarity is initialised as 0. The first string is stored in a variable str1 and the second variable is stored in a variable str2. The length of str1 and str2 is

stored in str1length and str2length. Then a new variable named similarity of strings is created and initialised as 0. If the length of both of the strings are less or equals to 0, then the similarityOfStrings is also set as 0. If their lengths are equal or they are synonyms or one of them contains the other, similarityOfStrings is set as 1. Then a variable distanceOfString is created to store the distance between two strings. The length of str1 and str2 are compared to determine the greater length which is then stored in a variable named maxLength. The distanceOfString is then divided by maxLength which is subtracted from 1. The result is then squared and stored in the variable called similarityOfString which is then returned. Thus, part A of the algorithm is concluded which determines the similarity between two strings.

Part B of the Algorithm determines the maximum node similarity in the list. Variables nodelst1 and nodelst2 are created to store the first and second list of unit sentence node. Two variables, maxSimilarity and similarityOfNode are created and initialised to 0. These two variables will later be used to store the maximum similarity between nodes and the similarity for each of the nodes by comparing them with nodes from the other list. Then a node from nodelst1 is taken and compared with each of the nodes from nodelst2 to determine the similarity between the pair using the steps from part A. This process is done for each of the members of nodelst1 which means all the members of nodelst1 are paired with all of the members of nodelst2 once and the similarity of each pair is determined. Then the maximum similarity is calculated and stored in maxSimilarity variable which is then returned as a conclusion of part B of the algorithm.

Part C of the algorithm gets similarity between nodes in list. First of all, similarity distance between node1 and node2 is determined and stored in a variable named nodeSimilarity. If it is greater than 1, 0 is returned. Otherwise, nodeSimilarity is multiplied with 2/3 and returned. These steps are carried out to make the nodeSimilarity more consistent and facilitate appropriate formatting.

Part D calculates the similarity between the ancestor nodes of the current node. Two-unit sentence nodes in lists are stored in firstNode and secondNode variables. The ancestor nodes of them are stored in variables AncNode1 and AncNode2. Then part B is used to get the maximum

similarity between these two ancestor nodes.

Similar process is done in part E to determine the descendant node similarity. desNode1 and desNode2 are used to store the descendants of the two specified nodes which are then compared using part B to calculate the maximum node similarity between desNode1 and desNode2.

Thus, similarity distance between sentences is edited in five parts by comparing the nodes themselves as well as their ancestors and descendants.

5.3 Process Map Construction

5.3.1 Failure Report Analysis System

The Failure report analysis system for the process map construction is composed as follows in Figure 5.3. The functions of each module are as follows.

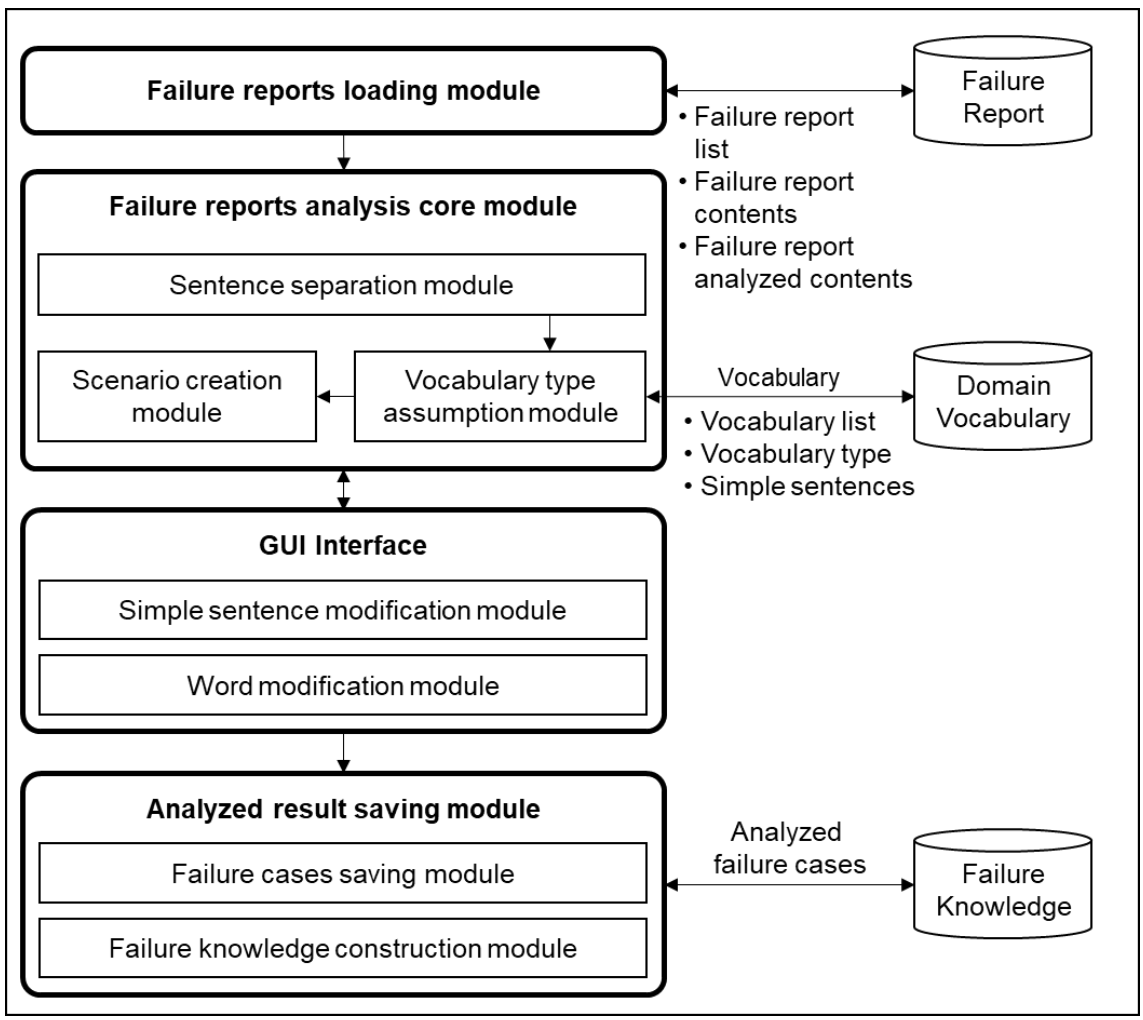


Figure 5.3 Architecture of Failure Report Analysis System

Failure Report loading module:

- Output a list of Failure reports created outside the system
- Check the contents of the failure reports.
- Batch processing of multiple Failure reports
- Identify case information for preprocessed reports

Failure Report Analysis Core module:

- Create rule-based fault case based on the Title / Phenomenon / Cause / Action of the Failure report.
  - Rule-based Failure report statement separation
  - Normalization of the vocabulary (elimination of postposition / adverb) and extraction of each vocabulary type based on the contents stored in the process map (DB)
  - Estimate the type of non-existing vocabularies
  - Create a final failure case scenario based on the estimated vocabularies
    - ✓ Type-based segregation
    - ✓ Restore lost items (Restore target and status)
    - ✓ Remove duplicate short texts

GUI Interface:

- Short text level
  - Order adjustment function of failure scenario created by analysis module
  - Delete insignificant short texts
- Vocabulary Level
  - Correction functions of vocabulary-type errors estimated by the analysis module.
  - Correction function of vocabulary type estimated by analysis module
- Regenerate the case scenario based on the information modified by the user. (Request to analysis core module.)

Analyzed result saving module:

- Storing the newly constructed lexical information by the user
- Each short statement of the finally analyzed Failure report is stored in the process map as each fault (action) phenomenon
- Establish relationship information of each failure (action) phenomenon (cause result / action order)

5.3.2 Process of Failure Report Analysis

Failure reports written in natural language are transformed into structured forms using natural language processing techniques and then stored in the process map. The analysis is conducted automatically by the system and the results of that can be modified directly by the user through the user interface. The contents stored in the process map are used as information for analyzing the failure report, and then those analysed contents are connected to each other with the same relation so that the information can be easily accessed and expanded [147]. Figure 5.5 briefly shows the failure analysis process.

Figure 5.4 is an example of a failure report and an analyzed failure case. The failure report includes the name of the equipment where the failure occurred, the date and time of the failure, the failure condition, and the cause of the failure. The analyzed failure cases are represented in the list on the right side of the failure report, and they can be reconstructed in the order of the failure cause, failure phenomenon, and countermeasure method. It can also be used as causal knowledge. The blue letter is the object of the failure and the red letter indicates the status of the facility. In addition, by providing a user interface, field experts can directly edit the objects and phenomenon of the failure and adjust the order of the failure phenomenon.

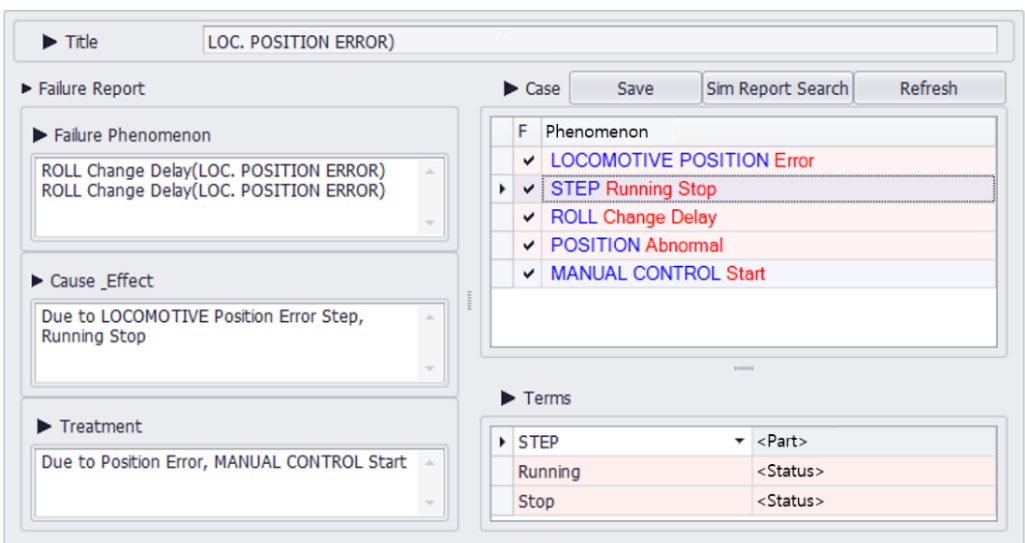


Figure 5.4 An example of Failure Report and Analyzed Result

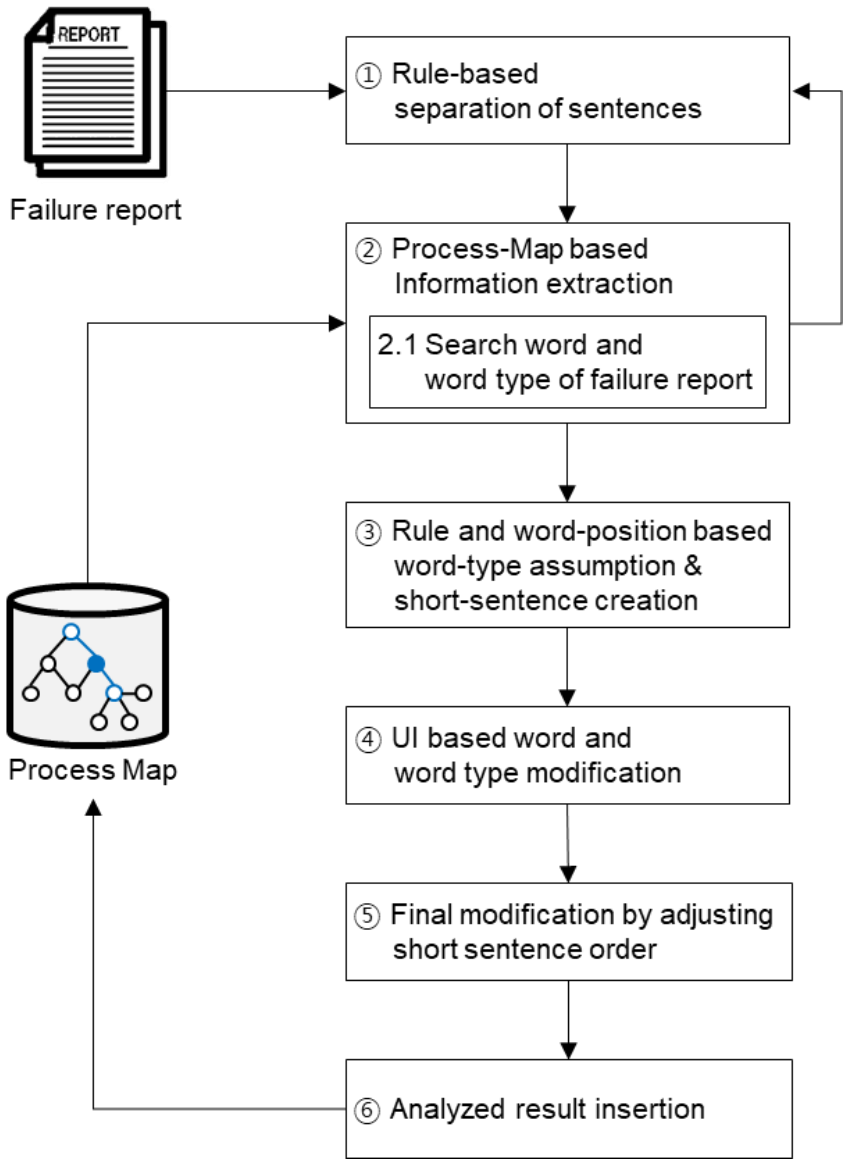


Figure 5.5 The proposed failure report analysis procedure

① **Rule-based separation of sentences**: This stage separates sentences into short clauses consisting of subject and predicate, which makes them the minimum units of a sentence. Sentences are separated using the regular expression based sentence separation rules.

② **Process-Map based Information extraction:** At this stage, the morpheme and part of speech from the short sentences separated in step 1 are extracted and at step 2.1., words are extracted from the process map as subject and predicate. After these two processes the type of that specific word is defined. If there are no exact matches with the words in the process map, partially matched words are retrieved. After that, if the extracted string from the process map is matched with the word in the short sentence, or if the string matches the search, the corresponding string marks in the short sentence and the string is registered as the candidate of the corresponding word type. Otherwise, if the string extracted from the process map does not completely coincide with the word within the short word, the corresponding string is marked and registered as a subordinate of the candidate word type.

③ **Rule and word-position based word-type assumption & short-sentence creation**: If the type of vocabulary recognized in step 2 corresponds to the phenomenon (function-breakdown-action), it is partitioned into separate sentences based on the vocabulary. If the vocabulary is not recognized in step 2, the type estimation of the unregistered word is performed. If the vocabulary is not an abbreviation and the position of the word is at the end of the sentence, it is estimated as a status term. If there are duplicated semantic meanings, the similarity degree between short sentences is calculated. If the predefined threshold is exceeded, the one with highest similarity is selected, and the short sentences with lower similarities are removed. If the word corresponding to the object or phenomenon is omitted from the natural language itself, the word restoration is performed in three steps: 1) If the word corresponding to the object of the short sentence is omitted and there is a restorable object from the

previous sentence, then the object of the short sentence is restored by using the object of the previous sentence. 2) If the target word of the short sentence is omitted, but the rest of the sentences cannot be restored, then the subject of the short sentences is restored using the subject of the next sentence. 3) If the status expression of the short sentence is omitted, then the state word from the following sentence is restated.

④ **UI based word and word type modification**: The analysis of the failure report is performed automatically up to the step 3, and at present stage, the user can directly supplement the contents through the UI. If the user wants to change the vocabulary type of the analyzed short sentences, they can change the vocabulary by searching them from the process map and the domain vocabulary using the interface.

⑤ **Final modification by adjusting short sentence order**: This allows the user to change the order of the short sentences which are automatically analyzed up to step 3 through the UI. The user can change the order of the selected short sentences up, down and remove the selected short sentences. In addition, the user can define new short sentences and insert them at the desired positions.

⑥ **Analyzed result insertion**: It is the step of storing the result of the analysis that is finally completed in the process map. This is done after sorting the order of short paragraphs in the order of Phenomena-Cause-Actions and then save them in the process map. In the case of a failure-related short sentence, the object and status are inserted into the failure phenomenon part of the process map, and the cause and effect relationship between the input failure phenomena is established. Also, in case of a short sentence for an action, the object and phenomenon of the short sentence are entered in the action part of the process map, and the precedence relationship between the phenomena for the action is set up.

5.3.3 Methodologies of Failure Report Analysis

1) Sentence Separation

Because the proposed Failure report analysis system was developed using the Korean failure report, I propose a method based on the Korean natural language processing. Failure reports and information professionals create their own, and there are a variety of expression forms. Statement consists of a minimum of the short form with subject and predicate, and is extended through the relationship between short texts.

This thesis is divided into the 'Status' corresponding to the predicate and the 'Part' corresponding to the 'subject.' It aims to extract an element of these two sentences. 'Part' generally may be in the equipment / parts, whereas the 'Status' means the status / operation of a 'Part.'

In the first analysis stage of the failure report, a morphological sentence separation method using a regular expression is used. As shown Table 5.1, The separation criterion becomes an investigation and conjunction, for example, '하고\b' ('hago' = 'and') separates sentences based on conjunctions that connect the two sentences. This process works automatically.

Table 5.1 Examples of rule separation using regular expressions

\r\n	\b 시[에]?\b *si*[*e*], time
(으로|로){1}\s?(의|인){1}(하여|해|한){1}\b (*eulo*|*lo*) (*ui*|*in*) (*hayeo*|*hae*|*han*), (to/due/because)	상태에(서)?\b *sang-tae-e-*(*seo*), instate
(하게){1}[\s]*(되|하){1}(며|어|면|였|었){1}[^\s]*\b (*hage*) (*doe*|*ha*) (*myeo*|*eo*|*myeon*|*yeoss*|*eoss*), with / to / to be	
(하게){1}[\s]*(되|하){1}(며|어|면|였|었){1}(){1}(있는){1}[^\s]*\b (*hage*) (*doe*|*ha*) (*myeo*|*eo*|*myeon*|*yeoss*|*eoss*) (*issneun*), with / to / to be / being	
\b[이]?후[에]?[도]?\b [*i*] *hu*[*-e*] [*-do*], ever after	(으로){1}\b (*eu-lo)*, to
한\b *han*, -ed	됨\b *do-em*, being
된\b *doen*, -ed	하고\b *ha-go, and*
난\b *nan*, -ed	그(리고|래서|러므로|런데){1}\b *geu*(*li-go*, and)|(*-lae-seo*, so)| (*-leo-meu-lo*, therefore)| (*-leon-de*, by the way)
\b 중[에]?\b *Jung* [*-e*], during	따른\b (*dda-run*), depended
과정[^\s]*\b *gwa-jeong*, process	(때문에){1}\b (*ttaemun-e*), due to
(하|되|시키){1}면서 (*ha-myeon-seo*, while doing | *doe-myeon-seo*, as soon as | *si-ki-myeon-seo*, and)	(에서){1}\b (*eseo*), in
되어 있는(데)?\b *do-eeo iss-neun-(de)*, became	\b 때[에]?\b *tta[ee]*, when
(되어|되){1}\b (*do-eeo*|*doe*), became	결과\b *gyeolgwa*, result
(하여|해){1}\b (*ha-yeo*|*hae*), so	(을|를){1} (위){1}(하여|해|한)?\b (*Eul*, of) |(*leul*, to) (*wi-hayeo*, for) |(*wi-hae*, for)|(*wi-han*, for)
\b[이]?전[에]?[도]?\b [*i*] *–jeon* -[*e*] [*do*], before	

Algorithm 4 shows the sentence separation procedure. A string named report is used to store the failure reports. An array named sectioned report is created which stores the separated cause, status and act sections of the report. Then sentences are split and stored in an array called splitSentences. An array called candidate stores the candidate words which are searched from the process map. Processed map is stored in an array called PM and marked sentences with the words from process map is stored in an array called markedSentences. An array called finalSimple is created to store final simple sentences and candidates to store strings in process map. A function named separateSentences is created which takes report as a parameter and separates the sentences. A makeSections function is used on report to make sections from the report which are then stored in the sectionedReport array. For all the sections in sectionedReport, sentences are split, and candidates are retrieved by using the process map. Each of the sentences and candidates are passed as parameters in the similarity function. If they are found to be similar, then they are marked using a mark function and put in a variable called marked. These marked variables are pushed on a stack of markedSentences.

Algorithm 4 : Sentence separation

```
1:  LET String report ← failure report
2:  LET Array sectionedReport ← seperated report into cause, status and act sections
3:  LET Array splitSetences ← split sentences
4:  LET Array candidate ← candidate words searched from process map
5:  LET Array PM ← process map
6:  LET Array markedSentences ← marked sentences with the words from process map
7:  LET Array finalSimple ← final simple sentences
8:  LET Array candidates ← strings in process map
9:
10: FUNCTION seperateSentence(report)
11: sectionedReport ← makeSections(report)
12: for all section sec in sectionedReport do
13:       splitSetences ← simpleSplit(sec)
14:       candidates ← retrieveCandidate(PM)
15:       for all string sentence in splitSetences do
16:             for all string str in candidates do
17:                   if similarity(sentence, str) = TURE then
```

```
18:                         marked ← mark(sentence, str)
19:                     end if
20:                     push(markedSentences, marked)
21:                 end for
22:             end for
23:             for all string sent in markedSentences do
24:                 sent ← splitComposition(sent)
25:             end for
26:             finalSimple ← extractDuplicate(markedSentences)
27:             finalSimple ← propagateFromPrevious(markedSentences)
28:     end for
29:     RETURN finalSimple
```

All strings in marked sentences are passed in a splitComposition function to split their composition and the returned string is stored in a variable called sent. The duplicated are extracted from marked sentences and stored in the finalSimple array. Strings are also propagated from previous ones and are also stored in finalSimple which is then returned.

2) Rule based word normalization and stop word removal

As described above, the proposed system extracts 'Part' and 'Status,' the minimum units from the failure report. The other sentence components are unnecessary, and since the importance is low, only the sentence is extracted through the pre-processing process. With that, the sentence is simplified. A sentence can have the same meaning in various expressions. The normalization process is performed by replacing similar expressions with common vocabulary to make the meaning of the sentence simple and consistent. For example, as shown in Table 5.2, an expression such as '되(지)(않|안|못)(아|되|됨|하|함|했)' is replaced by 'Disable(불능: Bu;reung).' In addition, the stop-word, which is an unnecessary element, is removed. For example, the bullet symbol (eg.1 and 2) or the element like DATE/TIME are removed.

Table 5.2 Examples of stemming and stopword removal using regular expressions

Rules	Result
되(지){1}(않|안|못){1}(아|되|됨|하|함|했){1}[^\s]*\b ***doe*(*ji*) (*anh*|*an*|*mos*) (*a*|*doe*|*doem*|*ha*|*ham*|*haess*), not	불능 (***bul-neung***, Inability)
(않|안|못){1}(아|되|됨|하|함|했){1}[^\s]*\b (*anh*|*an*|*mos*) (*a*|*doe*|*doem*|*ha*|*ham*|*haess*), not	불능 (***bul-neung***, Inability)
일어[^\s]*\b ***il-eo***, occur	
\d{2,4}[/.]{1}\d{1,2}[/.]{1}\d{1,2}[^\s\w]*\b	![DATE]!
\d{1,2}월[ ]*\d{1,2}일[^\s\w]*\b (***weol***, month) (***el***, day)	![DATE]!
\d{1,2}시[ ]*\d{1,2}분[^\s\w]*\b (***si***, time) (***bun***, minute)	![TIME]!
\d{1,2}:\d{1,2}[^\s\w]*\b	![TIME]!
\([ ]*!\[TIME\]![ ]*\)	![TIME]!
!\[TIME\]![\t]*(-|~)+[\t]*(!\[TIME\]!){0,1}	![PERIOD]!
[\d]+/[\d]+[ ]*([월화수목금토일]+)[^\s\w]*\b [***wol-hwa-su-mog-geum-to-il***], [Monday Tuesday Wednesday Thursday Friday Saturday Sunday]	![DAY_DATE]!
!\[DATE\]!	
!\[TIME\]!	
!\[PERIOD\]!	
!\[DAY_DATE\]!	
\([^\)]*\)	
^:[^\w]*\b	
^\d{1,2}[/.\)]{1}	

The result through the 1), 2) process as Table 5.3 shown. Original sentences 'Broken due to the fatigue load of FLAPPER CYLINDER' 'in the' Cause 'item of the table are separated by postposition ('Uihan (의한)': by). An example of stop-word processing can be found in the 'Treatment' section. Sentence 1.12: 30 ~ 'is removed from' 1.12: 30 ~ FLAPPER malfunctioning radio reception. ', And' FLAPPER malfunctioning 'and' operational radio reception 'are separated by sentence separation rule.

Table 5.3 Examples of the output of stemming and stop-word removal

Failure phenomenon	
Failure Report	HSB DELIVERY FLAPPER CYLINDER ROD **작업중 절손.** **(*jag-eobjung jeolson*, *Cutting loss during work*)**
	HSB DELIVERY FLAPPER CYLINDER ROD **절손 (*jeolson*, *Cutting loss*)**
Applied	HSB DELIVERY FLAPPER CYLINDER ROD **작업중 절손**. **(*jag-eobjung jeolson*, *Cutting loss during work*)**
	HSB DELIVERY FLAPPER CYLINDER ROD **절손 (*jeolson*, *Cutting loss*)**
Cause	
Failure Report	FLAPPER CYLINDER **피로하중에 의한 절손.** ***(pilohajung-e uihan jeolson*, *Fatigue-induced cutting loss)***
Applied	FLAPPER CYLINDER **피로하중에 의한** ***(pilohajung-e uihan*, *Fatigue-induced)***
	절손. (*jeolson*, *Cutting loss*)
Treatment	
Failure Report	1. 12:30 ~ FLAPPER **작동불량으로 조업 무선수신**. **(*jagdongbullyang-eulo jo-eob museonsusin*,** *due to malfunction operational radio reception)*
	2. 12:33 ~ 12:37 **현장도착 및 원인파악.** **(*hyeonjangdochag mich won-inpaag*, *Site arrival and cause identification*)**
	3. 12:38 ~ 12:45 ROD **절손으로 용접**. ***(jeolson-eulo yongjeob*, *Welding by cutting)***
	4. 12:46 ~ 07:49 **운전실 작동** TEST **정상작동 조업실시**. **(*unjeonsil jagdong* TEST *jeongsangjagdong jo-eobsilsi,*** *operation room working* TEST *Conduct normal operation)*
Applied	FLAPPER **작동불량으로** **(*jagdongbullyang-eulo*, *Due to malfunction*)**
	조업 무선수신. **(*jo-eob museonsusin*, *Operational radio reception*)**
	현장도착 및 원인파악. **(*hyeonjangdochag mich won-inpaag*, *Site arrival and cause identification*)**
	ROD **절손으로 (*jeolson-eulo*, *by cutting*)**
	용접. (*yongjeob*, *Welding*)
	운전실 작동 TEST **정상작동 조업실시**. **(*unjeonsil jagdong* *TEST* *jeongsangjagdong jo-eobsilsi,*** *operation room working TEST Conduct normal operation)*

In order to distinguish 'Part' and 'Status' from the separated paragraphs through process (1) and (2), it is necessary to extract the vocabulary types. To do this, a domain terminology dictionary is used. The term dictionary is constructed through a domain dictionary, a manual, etc. using terms / facility names used in the domain.

First, the vocabulary registered in the term dictionary DB is searched and the vocabulary type is extracted.

- If the search result matches the search result exactly, the search result is used as is.
- If it partially matches the search result, DB information is used according to the condition.

Second, specify the search target.

- Only the vocabulary stored in the term dictionary is searched by the standard vocabulary in advance.
- Due to the existing report analysis results, unprocessed and stored vocabulary information does not exist. However, it can be used if it is registered as a standard vocabulary later in the post-processing.

3) Vocabulary Search

In order to retrieve a vocabulary from a terminology dictionary, the following process is necessary:

First, all the words included in the sentence (phenomenon / cause / action) of each part of the failure report are retrieved from the DB.

- Extract vocabularies based on INSTR query.

Second, tag the meaningful vocabulary in each sentence among the extracted vocabularies.

- Extracts combinations of vocabulary and compound words existing in a word in a sentence.
- If the vocabulary application order is longer in length, more precedence is given to what is located in front of the characters in the subject.

Third, separation of compound words

- When multiple vocabularies are tagged in a single word, it is assumed that the vocabulary is compounded and each vocabulary is separated.

Search results, when a partial match undergo the following processes such as

- Separate the vocabulary into three levels
  - Keyword: The part of the vocabulary found in the DB
  - Prefix: The part before the keyword
  - Suffix: The part that follows the keyword

- If the keyword part exists in the DB and the prefix / suffix exists in the predefined item, the keyword part is used as is.

4) Unregistered Word Normalization

The process for normalizing a vocabulary that is not registered in the DB and processing the keyword is performed as follows:

- If there is a key word vocabulary that exists at the same time in the prefix / suffix list defined in the prefix / keyword / suffix combination that can be combined for each vocabulary, the vocabulary is assumed as the key word.
  - To improve speed, I first combine vocabularies based on morphological analysis results.
  - If there are no satisfying keywords in the primary source, combine all possible cases by syllable.
  - If there is no combination of the above conditions, the entire vocabulary is assumed to be a key word.

- Separation of compound words
  - If one vocabulary is judged to be a combination of vocabularies existing in the DB,

the vocabulary is separated.

Table 5.4 is an example of a suffix list related to general postposition and conjunction.

Table 5.4 The list of proposition and conjunction

로 (*lo, to*)	으로 (*eulo*, to)	된 (*doen*, -ed)
에서 (*eseo, from*)	로 인하여 (*lo inhayeo*, due to)	중 (*jung*, medium)
인한 (*inhan, by*)	인하여 (*inhayeo*, due)	후 (*hu*, after)
의하여 (*uihayeo, by*)	때문 (*ttaemun*, because)	전 (*jeon*, before)
하였고 (*hayeossgo, and*)	에서 (*eseo*, in)	하여 (*hayeo*, so)
•••		

Table 5.5 is an example of a positional vocabulary related suffix.

Table 5.5 The list of suffix that represents positions and locations

Location Noun					
위 (*wi*, top)	**중간** (*jung-gan*, middle)	**아랫** (*alaes*, bottom)	**오른** (*oleum*, right)	**바깥** (*Bakkat*, outside)	**중** (*Jung*, medium)
윗 (*wis*, top)	**가운데** (*Gaunde*, middle)	**하** (*ha*, bottom)	**우** (*U*, right)	**외** (*Oe*, except)	**아래** (*Area*, bottom)
상 (*Sang*, top)	-	**밑** (*mit*, bottom)	**안** (*An*, in)	**주변** (*Jubyeon*, around)	**좌** (*Jwa*, left)
-	-	**왼** (*oen*, left)	**내** (*Nae*, inside)	**밖** (*Bakk*, out)	-
Location Aux					
쪽 (*jjog*, side)	**측** (*cheug*, side)	**방향** (*Banghyang*, direction)	**부** (*bu*, part)	**쪽편** (*Jjogpyeon*, side)	
편 (*pyeon*, side)	**단** (*dan*, only)	**부분** (*Bubun*, part)	**부단** (*budan*, part)	**쪽측** (*Jjogcheug*, side)	

Table 5.6 is an example of a prefix / suffix that is related to failure phenomenon.

Table 5.6 The list of prefix/suffix that is associated with failure phenomenon

Failure Prefix				**Failure Suffix**	
과 (*gwa*)	**무** *(mu)*	**안** *(an)*	**가** *(ga)*	**불** *(bul, impossible)*	**안됨** (*andoem*, no)
고 (*go*)	**역** *(yuck)*	**불** *(bul)*	**재** *(jae)*	**불** *(bul, impossible)*	**안함** (*anham*, do not)
미 (*mi*)	**저** *(jeo)*			**불가** *(bulga, impossible)*	

Table 5.7 is an example of suffixes related to use.

Table 5.7 The list of suffix that represents usage

Failure Suffix
용 (*yong*, for)
용도 (*yongdo*, purpose)
대 (*dae*, versus)

5) Suffix based Sentence Separation

If the vocabulary is included in the list of suffixes or vocabulary itself included in the list of suffixes defined in the predefined list (Table 5.8), it shall be divided into short sentences. Table 5.9 shows the vocabulary extracted from the characters through DB INSTR Query.

Table 5.8 The list of suffix for sentence separation

.	,	로 (*lo*, in)
으로 (*eulo*, to)	로인하여 (*loinhayeo*, due to)	으로인하여 (*euloinhayeo*, due to)
로의한 (*louihan*, by)	으로인한 (*euloinhan*, due to)	로의하여 (*louihayeo*, by)
으로의하여 (*eulouihayeo*, by)	되 (*doe*, become)	된 (*doen*, -ed)
전 (*jeon*, before)	후 (*hu*. after)	시 (*si*, when)
때 (*ttae*, time)	때문 (*ttaemun*, due to)	때문에 (*ttaemun-e*, due to)
하여서 (*hayeoseo*, to)	하였고 (*hayeossgo*, and)	인한 (*inhan*, by)
인해 (*inhae*, by)	인해서 (*inhaeseo*, by the way)	의해서 (*uihaeseo*, by)
•••		

Table 5.9 A sample output of term extraction using INSTR Query

Failure phenomenon	
Failure report	HSB DELIVERY FLAPPER CYLINDER ROD **작업중 절손.** **(*jag-eobjung jeolson*, *Cutting loss during work*)**
	HSB DELIVERY FLAPPER CYLINDER ROD **절손** **(*jeolson*, *Cutting loss*)**
Extracted words based INSTR Query	FLAPPER, ROD, CYLINDER, FLAP, BD, **절손, 작업** **(*jeolson*, *Cutting loss*), (*jakup*, working)**
Cause	
Failure report	FLAPPER CYLINDER **피로하중에 의한 절손.** (***pilohajung-e uihan jeolson***, *Fatigue-induced cutting loss*)
Extracted words based INSTR Query	FLAPPER, CYLINDER, FLAP, **피로, 절손** (***pilo***, *Fatigue*), (***jeolson***, *cutting loss*)
Treatment	
Failure report	1.12:30 ~ FLAPPER **작동불량으로 조업 무선수신**. (***jagdongbullyang-eulo jo-eob museonsusin***, *due to malfunction operational radio reception)*
	2.12:33 ~ 12:37 **현장도착 및 원인파악**. (***hyeonjangdochag mich won-inpaag***, *Site arrival and cause identification*)
	3.12:38 ~ 12:45 ROD **절손으로 용접.** (***jeolson-eulo yongjeob***, *welding by cutting)*
	4.12:46 ~ 07:49 **운전실 작동** TEST **정상작동 조업실시.** **(*unjeonsil jagdong* TEST *jeongsangjagdong jo-eobsilsi,*** *operation room working* TEST *Conduct normal operation)*
Extracted words based INSTR Query	ES, FLAPPER, **용접** (*yongjeob*, welding), ROD, **동조** (*dongjo*, tunning), **불량** (*bullyang*, bad), **작동** (*jagdong*, working), TEST, **조**(*jo*), FLAP, **운전실** (*unjeonsil*, operating room) **수신** (*susin*, reception), **파악** (*paag*, grasp) **조업** (*jo-eob*, operation), **운전** (*unjeon*, driving), **절손** (*jeolson*, cutting loss), **실시** (*silsi*, execution)

Findings from the registered language tagging, unregistered language normalization and their short separation may be found in Table 5.10. A non-guessed tagged vocabulary is displayed in '[ ]'. In the treatment part, from the sentence "1.12: 30 ~ FLAPPER malfunctioning radio reception." The result formed by the lexical criterion through such processing becomes '[FLAPPER] [operation] [bad]' and '[operation] wireless [reception].'

Table 5.10 A sample output of tagging and normalization

Failure phenomenon	
Failure report	HSB DELIVERY FLAPPER CYLINDER ROD **작업중 절손**. (***jag-eobjung jeolson***, *Cutting loss during work*)
	HSB DELIVERY FLAPPER CYLINDER ROD **절손** (***jeolson***, cutting loss)
Tagging and Normalization	HSB DELIVERY [FLAPPER] [CYLINDER] [ROD] [**작업**] (**jag-eob**, working)
	[**절손**] (***jeolson***, cutting loss)
	HSB DELIVERY [FLAPPER] [CYLINDER] [ROD] [**절손**] (***jeolson***, cutting loss)
Cause	
Failure report	FLAPPER CYLINDER **피로하중에 의한 절손**. (***pilohajung-e uihan jeolson***, *Fatigue-induced cutting loss*)
Tagging and Normalization	[FLAPPER] [CYLINDER] [**피로**] **하중 의한** [**절손**] ([***pilo***] ***hajung-e uihan*** [***jeolson***], *Fatigue-induced cutting loss*)
Treatment	
Failure report	1. 12:30 ~ FLAPPER **작동불량으로 조업 무선수신**. (***jagdongbullyang-eulo jo-eob museonsusin***, *due to malfunction operational radio reception)*
	2. 12:33 ~ 12:37 **현장도착 및 원인파악**. (***hyeonjangdochag mich won-inpaag***, *Site arrival and cause identification*)
	3. 12:38 ~ 12:45 ROD **절손으로 용접**. (***jeolson-eulo yongjeob***, *welding by cutting)*
	4. 12:46 ~ 07:49 **운전실 작동** TEST **정상작동 조업실시**. (***unjeonsil jagdong*** *TEST* ***jeongsangjagdong jo-eobsilsi,*** *operation room working TEST Conduct normal operation)*

Tagging and Normalization	[FLAPPER] [작동 (*jagdong*, working)] [불량 (*bullyang*, bad)] [조업(*jo-eob*,operation)] 무선(*museon*,radio) [수신(*susin*, reception)] 현장도착(*hyeonjangdochag, Site arrival*) 및 (*mich*, and) 원인(*won-in*, cause) [파악 (*paag*, grasp)] [ROD] [절손 (*jeolson*, cutting loss)] [용접 (*yongjeob*, welding)] [운전실(*unjeonsil*, operating room)] [작동(*jagdong*, working)] [TEST] 정상(*jeongsang*, normal) [작동(*jagdong*, working)] [조업 (*joup*, operation)] [실시(*silsi*, execution)]

6) Word Type Extraction and Assumption

If the result retrieved from the DB matches the vocabulary part or if the search vocabulary is an unregistered vocabulary, processing is performed. Table 5.11 shows examples of the results.

First, estimation of partial match vocabulary type

- First, I apply the type of keyword that exists in DB.
- If the prefix / suffix is applied to each classification defined before (use / location / fault); change the type to that type.

Second, estimation of unregistered vocabulary type

- I estimate the type based on the location of the vocabulary in the short text.
  - If this vocabulary exists in the end, it will be estimated as a phenomenon, otherwise it is an Object.
- If the prefix / suffix is applied to each classification defined before (use / location / fault), change the type to that type.

Table 5.11 A sample output of word type distinguish

1st Word type assumption		**Final Word type assumption**
[**용접**]**부** [***Yongjeob***]***bu***, [welding]side	[용접]부→**고장** (용접→**고장**) (***Gojang***, broken)	[용접]부→**위치** (***Wichi***, location)
HSB DELIVERY [FLAPPER] [CYLINDER] [ROD] [**절손**] [(***jeolson***, cutting loss)]	HSB →**대상** (***Daesang***, object)	HSB→**대상** (***Daesang***, object)
[FLAPPER] [CYLINDER] [피로] 하중 ([***pilo***] ***hajung***, *Fatigue-induced*)	**하중→고장** (***hajung***, weight)	하중→**고장** (***Gojang***, broken)

Algorithm 5 : Word type assumption

```
1:  LET Array finalSimple ← final split simple sentences
2:
3:  FUNCTION tagWordType(finalSimple)
4:  for all sentence sent in finalSimple do
5:
6:      if sectionType(sent) = CAUSE or STATUS then
7:          nodeType = FAILURE
8:          else
9:              nodeType = REPAIRE
10:         end if
11:         for all word w in sent do
12:             inferWordType(w)
13:     end for
14: end for
15: propagateFromPrevious(finalSimple)
16: RETURN finalSimple
```

Algorithm 5 is used to determine word type assumptions. Then function tagWordType is created which takes finalSimple as a parameter. It tags the words of finalSimple with necessary attributes. Section type of each of the sentences of finalSimple is determined and if the type is either cause or status, node type is expressed as failure. In another word, this specifies that specific sentence as the cause of failure. Otherwise, the node type is labelled as repair. Each word in sent is passed to a function named inferWordType which determines the type of inference. finalSimple is then propagated from previous and returned.

7) Word Type based Sentence Separation

If the vocabulary type corresponds to a phenomenon (failure / action / function), separate the short sentences based on the vocabulary.

First, if the phenomenon (failure / action / function) vocabulary appears consecutively, separate all.

- Pad damaged (failure) Abrasion (failure) Pad damaged / Abrasion

Second, exceptions that do not separate the phenomena that appears consecutively

- When the type of continuous phenomenon vocabulary is different
  - Fault + Action / action + fault / function + fault / function + fault /...
- When one or more of the consecutive vocabularies is a vocabulary estimated by the system
  - [Fatigue] Load: Since the load is a type of failure estimated by the system, fatigue is determined as a failure type so that even if the same type appears consecutively, it does not separate.

8) Restore Missing Word Type of Simple Sentence

Failure reports that are created without a formatted structure may result in omitted subject or repeated vocabularies. In this case, it is necessary to reconstruct the items lost through the context of the sentence, because the necessary components may be deficient if they are separated into short sentences. The restoration process is divided into two parts.

First, the first short sentence restoration consists of the following steps.

- If a short sentence has only a representation of a subject or phenomenon due to a line break or rule-based sentence separation error, and these short sentences appear consecutively, the two short sentences are combined and integrated into one. An example is shown in Table 5.12.

Table 5.12 A sample result of initial sentence restoration

Failure report (with line feed error)	**Result**	**1st Restoration**
TRUCK	TRUCK[PART]	**TRUCK[PART] DAMAGE[FAILURE]**
DAMAGE	DAMAGE[FAILURE]	
TRUCK STOP	TRUCK[STATUS] STOP[FAILURE]	TRUCK[STATUS] STOP[FAILURE]

Second, the second short sentence restoration consists of the following steps.

- In the case of a short text in which no object exists, the object is extracted from the previous short text and restored (Figure 5.7).
- If the expression of the phenomenon is a combination of two or more types (ex. Failure + action) (Figure 5.8).
  - Restore the previous sentence using the phenomenon expression corresponding to the posterior
- Decide what to restore short text
  - Short texts generated in the same sentence on the failure report are targeted.

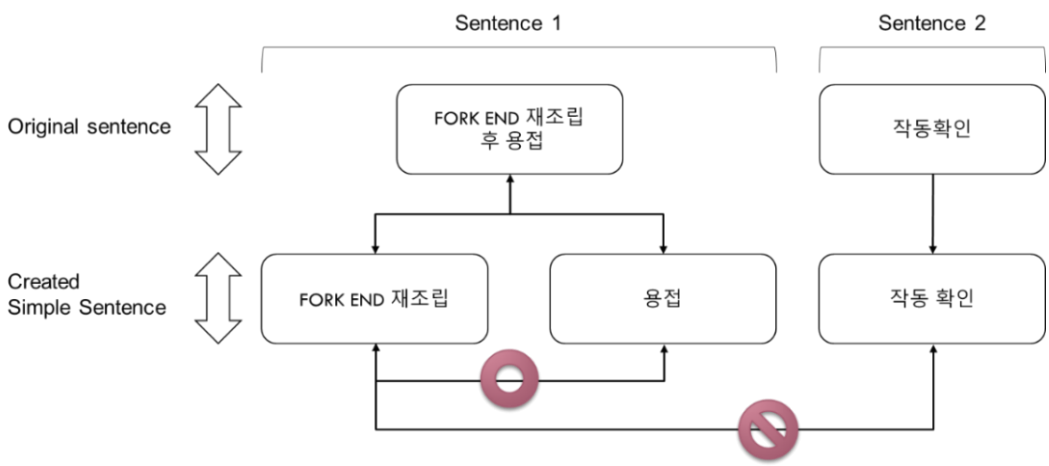


Figure 5.6 A conceptual diagram of sentence separation procedure

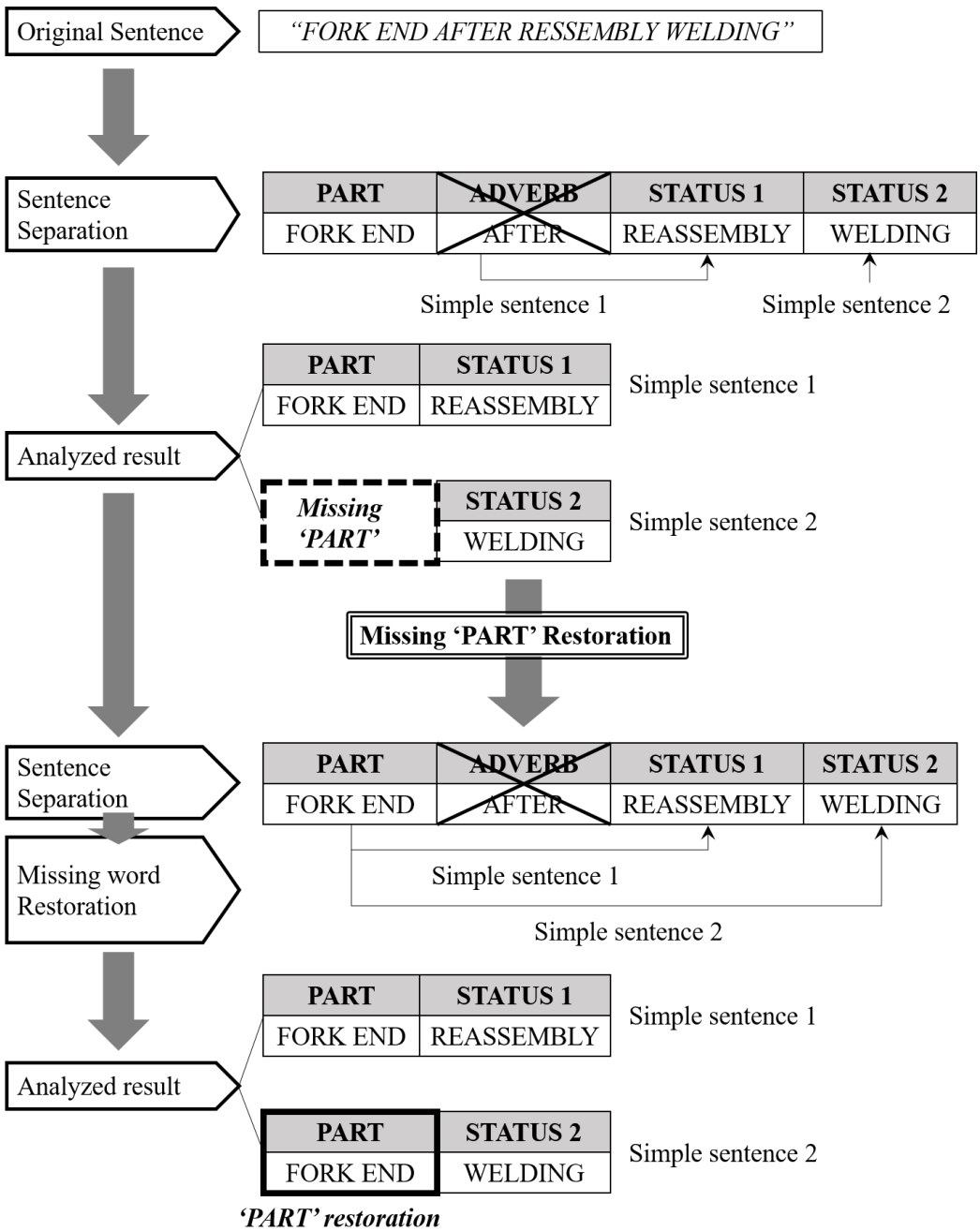


Figure 5.7 An example of category 'PART' restoration

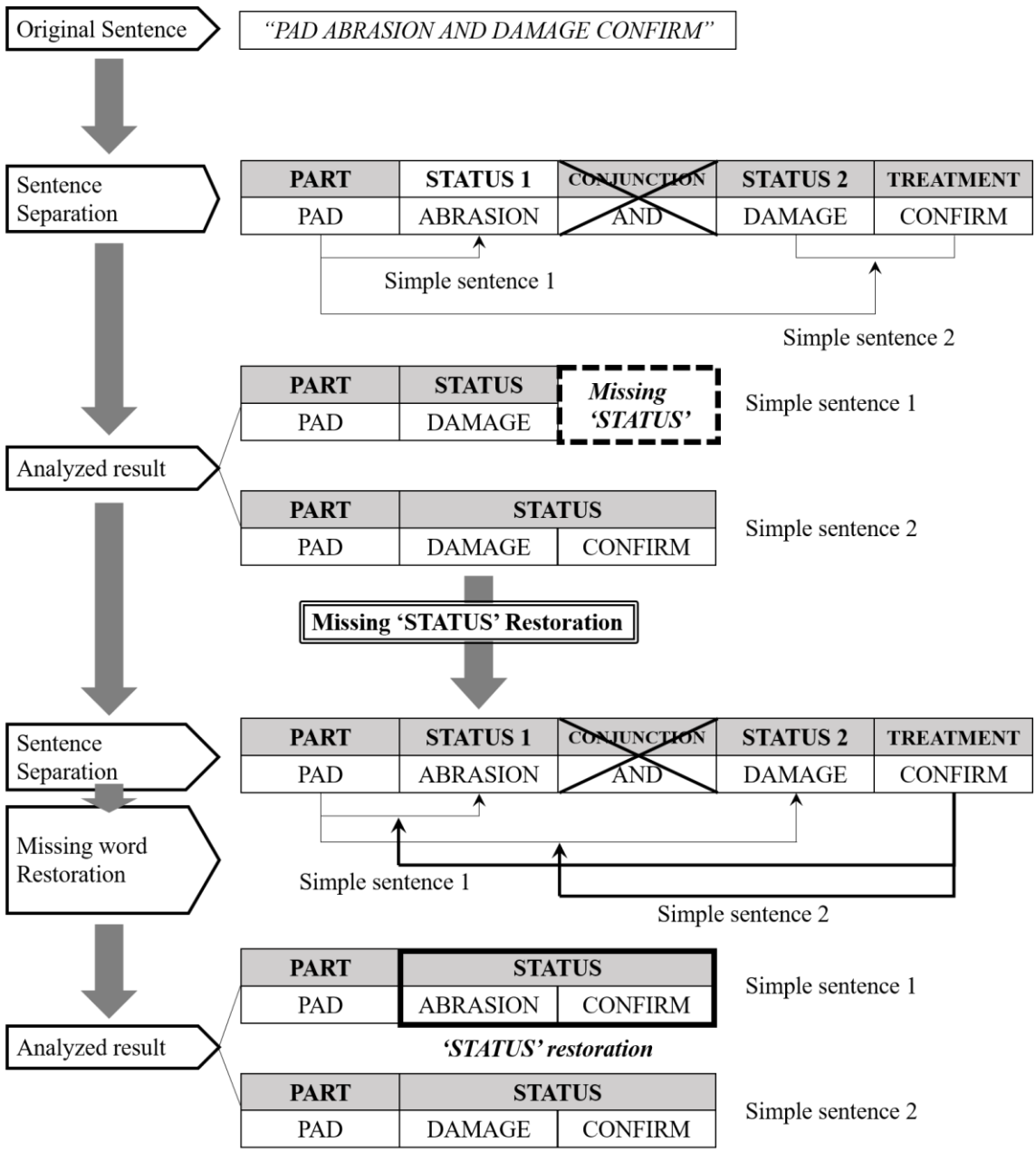


Figure 5.8 An example of category 'STATUS' restoration

5.3.4 Duplicated Simple Sentence Removal

There may be duplicate short sentences in the sentences, and in such cases, it is necessary to remove duplicate short sentences. This process is as follows:

- Calculate the similarity between short sentences and remove duplicate short sentences if duplicate short sentences exist.
  - Similarity (short sentence) = similarity (object) + similarity (phenomenon)
  - Similarity is calculated as the overlap between words.

- Consider the details of the short text.
  - Assume short paragraphs with longer lengths to be more detailed short paragraphs.

- If the details are the same,
  - Put weight on the first sentence and remove the sentence that appears afterwards.

Table 5.13 Examples of Duplicated Sentence Removal

Case Scenario	**Duplicated Simple Sentence Removal**	**Remarks**
FORK END DAMAGE	~~FORK END DAMAGE~~	‘Air Cylinder FORK END DAMAGE’ is more detail.
…	…	
…	…	
Air Cylinder FORK END DAMAGE	Air Cylinder FORK END DAMAGE	
FORK END DAMAGE	FORK END DAMAGE	Both sentences are completely in the same, So, following sentence is deleted.
…	…	
…	…	
FORK END DAMAGE	~~FORK END DAMAGE~~	

5.3.5 GUI for modification of analysed results

Case Scenario Short text level:

- Correct short texts of target/phenomenon
  - Modify the subject / phenomenon vocabulary of the short sentence to change the contents of the short sentence.
- Change order of short texts
  - Modify the content by changing the order of the short sentences that constitute the case scenario.
- Delete short texts
  - Delete short sentences that appear in the report but are meaningless to construct the case scenario.
- Change type of short statement (failure / action)
  - Select which type of phenomenon is included in each scenario in the case scenario.

Vocabulary Level:

- Modify the vocabulary that constitutes the short sentence
  - Changes the selected vocabulary to a form modified by the user.
  - If the modified vocabulary contains a space, the vocabulary is divided into a number of words separated by a space.
- Edit vocabulary type
  - Change the type of selected vocabulary to the type you choose: If there is the same vocabulary in the case scenario, change all vocabulary types at the same time. However, it does not change in case of the vocabulary in which the user previously selected the type directly.

- If the vocabulary is estimated from external short texts through the restoration function:
  - The vocabulary is displayed in italics on the user interface and cannot modify the information of the vocabulary.
  - If the information about the vocabulary is modified in the original short sentence with

the corresponding ocabulary, the other short sentences are reflected in the same way.

- If the vocabulary added by the restoration function is not correctly estimated, it can be deleted from the short text.

Calculate the similarity between the target vocabulary and the present vocabulary separately:

- Calculate similarity based on Edit-Distance
- Calculation of similarity based on DB analogy
  - If similarity relation is specified in DB, it is reflected as similarity 1

Failure scenarios:

- Flow of faults built by integrating Failure reports created by the same phenomenon in the same installation
- A failure scenario is made up of one or more failure cases.
- An installation can have one or more failure scenarios

Failure cases generated by the Failure report are being built in fault scenarios via the following two ways:

- Registered as an independent new failure scenario
- Integration with existing fault scenarios

The system proposes candidate scenarios that can be integrated with the case by sorting them based on the similarities of the fault phenomena included in the fault scenarios that have been established and the newly analyzed fault cases.

The similarity between all the fault phenomena belonging to the fault case and all the fault phenomena belonging to the scenario is measured based on the Edit-distance.

- The similarity of phenomena is measured the same as defined before

The similarity between the failure case and the scenario is defined as the sum of the largest

similarity values of each failure phenomenon belonging to the failure case.

Build a new scenario:

- Failure cases analyzed by the user are stored as scenario information as they are.
  - Built with the same content without any modifications

Integrate new cases into existing scenarios:

- Incorporate the currently analyzed failure cases into the flow of existing scenarios
  - The user creates new scenarios by adding the phenomena that appeared in the new failure cases in the middle of the existing scenarios

5.4 Evaluation

Through experiments, I demonstrate the suitability of the text-based similarity evaluation method used for comparing the similarity between cases in the process map.

5.4.1 Experiment data

To construct the process map, I used the failure report written in the actual plant domain. A total of 5,002 failure reports were collected during the period from October 2012 to July 2016, of which 713 failure reports were selected for the same plant. Of the 713 selected failure reports, 400 were used in the experiment except those containing the same content redundancy and poor content.

5.4.2 Experiment method

The experiment is carried out by the following two methods.

First, a fault phenomenon (part, status) with similarity was selected as a (pair) between failure reports. Second, I selected the most representative method for evaluating text similarity for experiments. The characteristics of each similarity evaluation result are analyzed by using Jaccard similarity and Consine similarity including Edit-Distance used in the proposed method.

5.4.3 Experiment Result

5.4.3.1 Text Similarity Analysis

Three similarity evaluation methods were applied to the collected fault phenomenon pairs, and the measured values were compared. The graph below shows the comparison of the measured values when each method is applied to the top 20 fault similarity pairs. As a result of the analysis, edit-distance showed the highest similarity value, but Jaccard and Consine also showed somewhat similar values. Also, it can be confirmed that the three methods are not significantly different in the aspect of the similarity measure value. Therefore, it can be concluded that there is no problem in evaluating the text similarity using any method. However, since the edit-distance used in the proposed method compares each of the characters constituting the word, it can be said that it is most suitable for a text comparison having a similar form. In the case of Jarccard, the similarity is evaluated by including the same character through the character intersection of two texts. Cosine similarity expresses text as a vector and evaluates the similarity as the distance between texts.

Failure reports are written by the field experts and may contain word spacing errors or typos. Also, when writing a sentence, the word form may change while the word and the postposition are used together. In this case, since the similarity can be evaluated more precisely by making a detailed comparison in the character unit constituting the word, edit-distance is judged as the most suitable method.

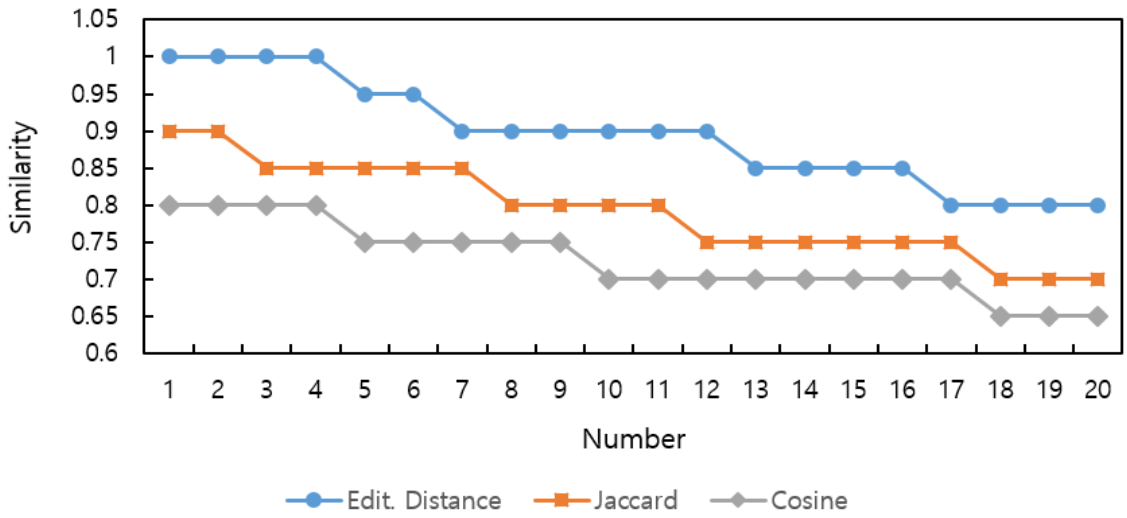


Figure 5.9 Similarity measure comparison based on top 20 similar node sets

5.5 Conclusion

In this thesis, I propose a method of constructing causal knowledge in network form by analyzing the failure report. In the present industrial field, I was worried about the use of failure reports in the form of a natural language for which the system is not available. The proposed method has been studied as a solution to this problem. I propose a natural language processing method to analyze the pattern of the failure report. The failure reports are automatically processed into formalized knowledge through the proposed natural language processing method. To construct knowledge in network form and to acquire appropriate knowledge, I constructed correlations between similarity-based knowledge. This not only enables you to quickly acquire relevant knowledge about the current problem but also allows experts to directly converge and edit similar knowledge into optimized knowledge. I have developed a tool to support this. Experimental results show that the similarity evaluation method applied to the proposed method is appropriate.

6 Hybrid Knowledge Representation Integration

This chapter introduces the expert knowledge described in chapter 3 and how to use the process map constructed as a failure case together in chapter 4. The fault prediction system based on this is proposed as shown in Figure 6.1.

In each section, how to use two types of knowledge together and check the performance through experiments is explained.

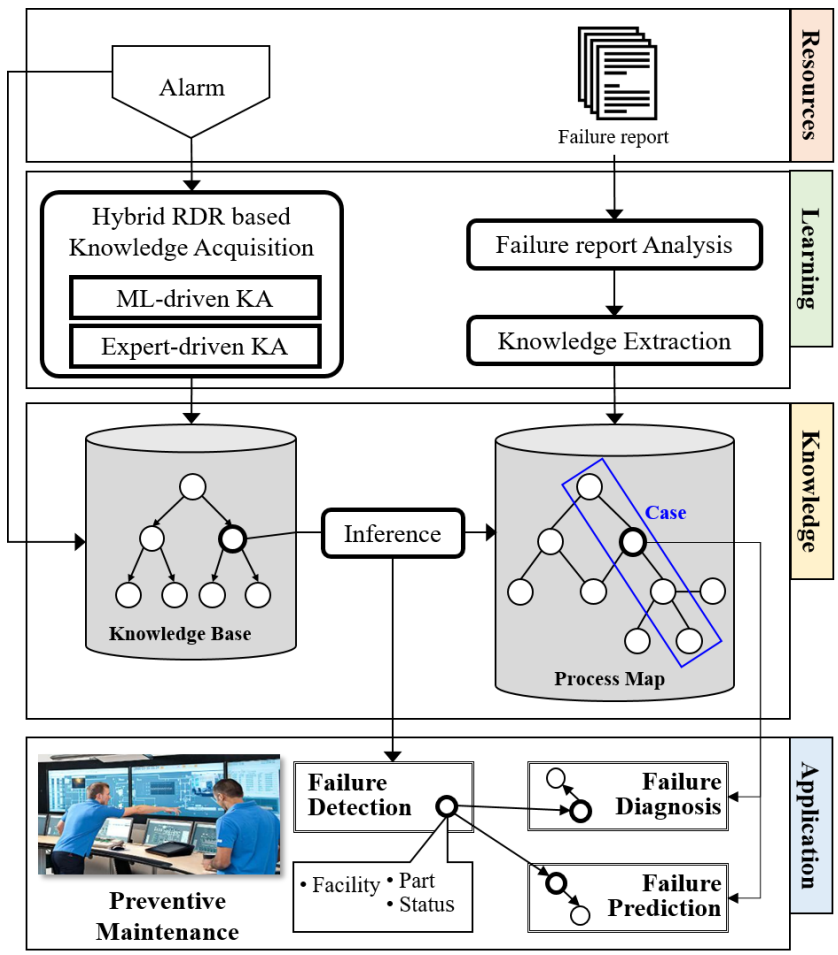


Figure 6.1 A conceptual diagram of the proposed failure prevention system

6.1 Knowledge Acquisition with Hybrid Knowledge Representation

In this section, the architecture of proposing system and the main components are described. The proposed system is composed of two closely interoperating main components, a rule-based expert system for processing alarms and a failure analysis system to predict future system failures. A knowledge base for finding the abnormal status of the facility and preventive or diagnosis knowledge base about abnormal signs are essential to perform preventive maintenance automatically with the system. Those two kinds of knowledge come from distinct sources, but they share equivalent unit knowledge for the failure diagnosis and prediction. The expert system generates and manages the knowledge for finding the failure signs. The failure report analysis system extracts knowledge from the failure reports, converts into causal relationship knowledge for failure diagnosis and prediction, and stores it into knowledge storage named as the process map.

The alarm knowledge is stored and managed in the knowledge base which is built by field experts to capture the problem of the system in real time. The expert system for alarm handling captures the failure indications from the alarm and retrieves the appropriate failure cases from the process map and provides them to field experts in predicting future failures or diagnosing current failures. The alarm knowledge constituting the system is generated after considering the relationship between failures and alarms using the user interface by field experts who have experienced a failure. Knowledge is modeled on an IF-THEN-based rule basis to represent human knowledge. The field experts generate and continuously manage the phenomena in the knowledge base which is generated by the alarm pattern. The knowledge acquisition engine (Figure 6.2) also works based on the experience of the experts. When an hourly alarm is collected, the inference engine finds the appropriate rule to process the alarm from the knowledge base and derives the result. The derived results are used to find failure cases matched in the process map.

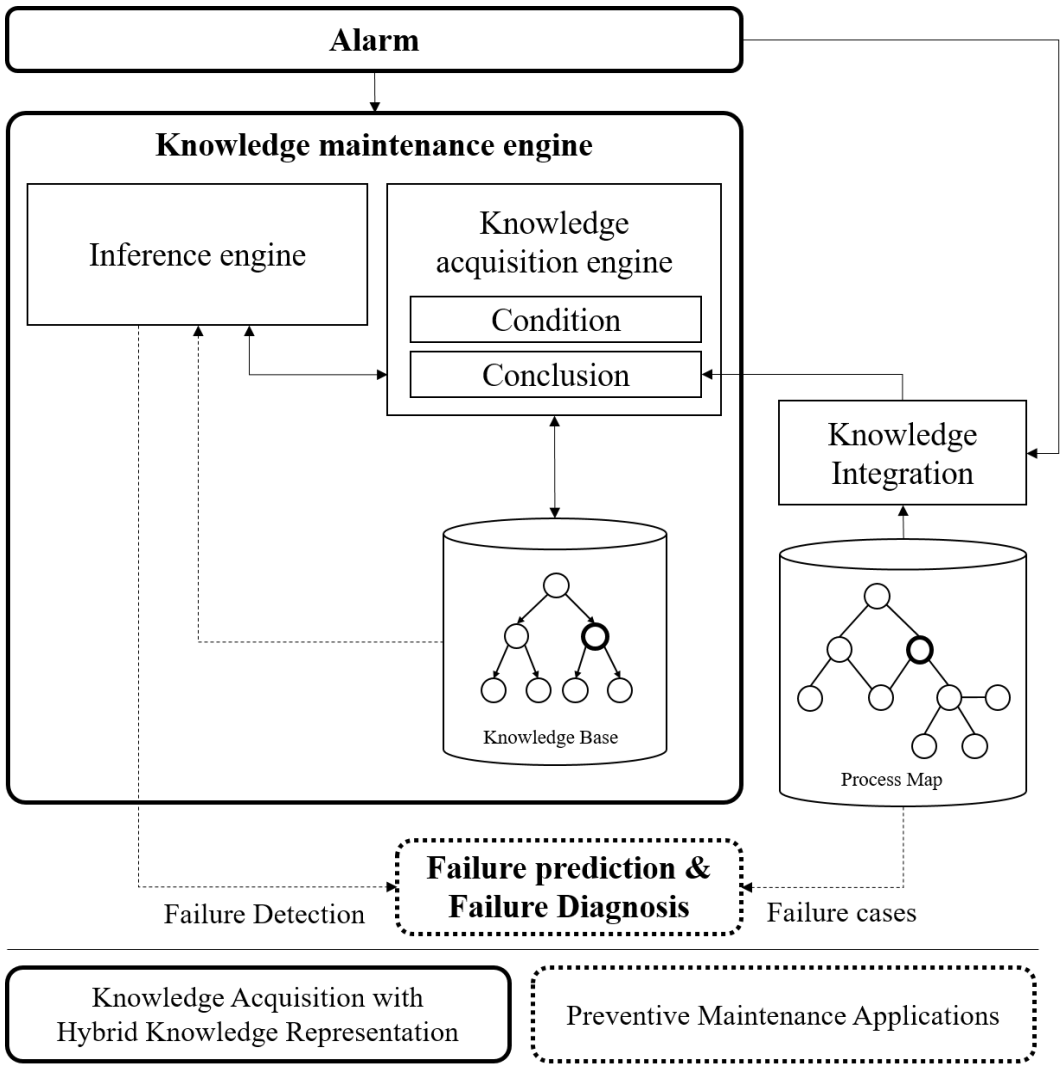


Figure 6.2 The knowledge acquisition process for the proposed failure prevention system

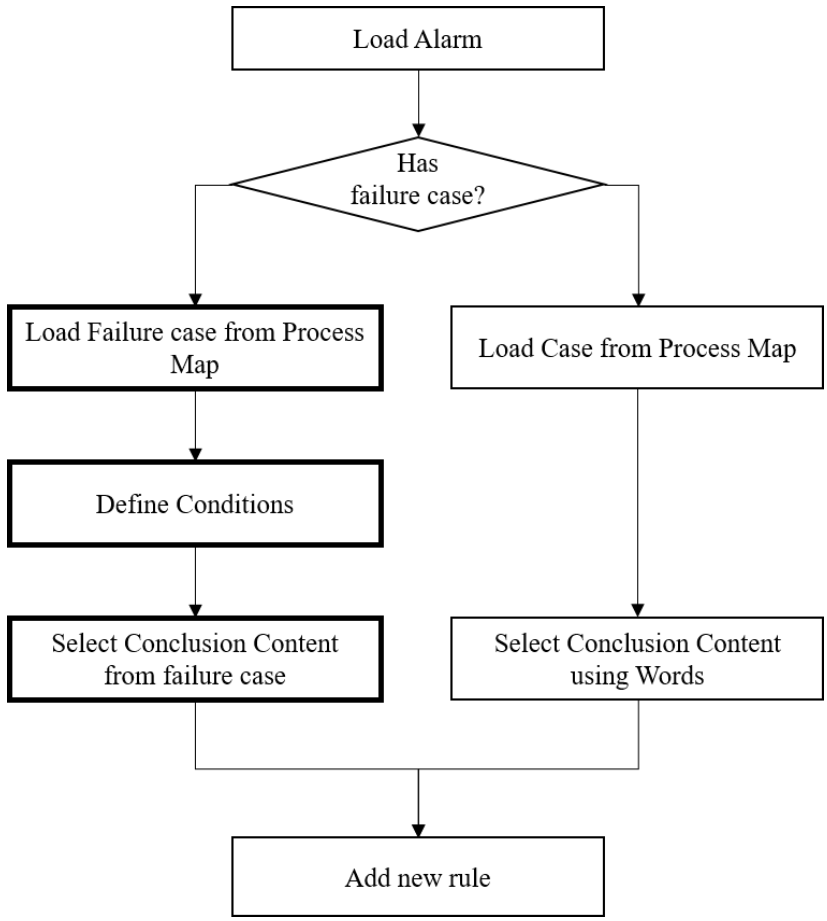


Figure 6.3 The procedural use case of conclusion retrieval using process map

The conclusion of the rule is the same as the failure phenomenon and includes Part / Status. Thus, both knowledge can be interlocked by matching the conclusion of the rule with the failure phenomenon of the process map. To do this, I use the failure report created at the time of fault occurrence in the conclusion definition process. That is, in the analyzed failure report, the failure phenomenon related to the alarm is selected as a conclusion of the rule, and the rule is defined. If there is no failure report, or before analysis, the user can enter the conclusion directly. The process for this is shown in Figure 6.3, and actual knowledge acquisition interface is shown in Figure 6.4.

Figure 6.4 The knowledge acquisition interface for human experts

Algorithm 6 : Expert System Structure

```
1:  LET Array Alarm_i ← Alarm list for the ith hour
2:  LET rule set KB ← rule-base knowledge base
3:  LET rule R_j ← the rule in the KB with the id j (R_0 is a root rule)
4:  LET String Con_j ← conclusion of R_j
5:  LET Array Result ← collection of satisfied rules
6:  LET Object PM ← process map
7:  LET case C_k ← the failure case in the PM
8:  LET String Phe_k ← the phenomenon of C_k
9:  LET Array Result_PM ← matched failure cases with inference result Result

10: FUNCTION Inference(Alarm_i, KB)
11: if satisfied(R_0, Alarm_i) then
12:     if R_0 has child rules then
13:         for all child rule Child do
14:             if satisfited(Child, Alarm_i) then
15:                 if hasChild(Child)=TRUE then
16:                     Inference(Child, Alarm_i)
17:                 else
18:                     push(Result, Child)
19:                     pop(Result, R_0)
20:                 end if
21:             end if
22:         end for
23:         if no child rule satisfied then
24:             push(Result, R_0)
25:         end if
26:     else
27:         push(Result, R_0)
28:     end if
29: end if

30: FUNCTION ResultMatching(Result, PM)
31: for all Con_j of R_j in Result do
32:     for all Phe_k in PM do
33:         if matched(Con_j , Phe_k) then
34:             push(Result_PM, Phe_k)
35:         end if
36:     end for
37: end for
```

The main structure of the expert system is illustrated in Algorithm 6. The alarm list for each hour $Alarm_i$ is used as the input for inference against the expert knowledge base *KB* (Line 13).

The inference starts from the root rule R_0. Once the current starting rule is satisfied, all the child rules will be evaluated. Only when there is no child rule for the current starting rule will the current starting rule become the inference result (Line 29). If there is the $Alarm_i$ can satisfy any child rule, the inference will continue with its child rules of this satisfied rule until no more child rules can be satisfied (Line 17 - 19). Otherwise, the current starting rule is stored as one of the inference-resulted rule (Line 27). The final result of the inference *Result* will be a collection of rules that are satisfied by the $Alarm_i$. The conclusion of the each satisfied rule is in the form of the phenomenon that is stored in the process map *PM*. These phenomena from the conclusion will be used to find the corresponding failure case in the *PM* that contains the same phenomenon (Line 35-38). The failure diagnosis and prediction in the failure case can then be utilized.

6.2 Alarm Knowledge Representation

Field experts can create rules by handling alarms collected through the knowledge acquisition engine. The attributes used in the condition part of the rule are as follows: 1) Facility name: The facility that generates alarms (e.g. Finishing Mill), 2) Alarm message: The textual representation of an alarm (e.g. F3 BOT PC APC ERROR), 3) Counts: The number of times the alarm occurred in every hour (e.g. 1), 4) Duration: The duration of the alarm in every hour (e.g. 96), 5) Rate: The ratio of the alarm to one day (e.g. 26.67)

The conclusion of the rule is the same structure as the unit knowledge base in the process map. Thus, the conclusion part consists of the facilities, objects, and phenomena. The field expert who generates the rule can define the conclusion by searching and selecting the failure phenomenon in the process map. Therefore, since the inference results of the failure prediction system are mapped to the failure knowledge, a process map can be used for failure prediction. Figure 6.5 shows an example of a knowledge base where each rule defines the rule condition using five attributes of the alarm, and the conclusion is defined using the phenomenon which constitutes the case of the process map. For example, Rule 1 is defined as a condition with the

number of occurrences (Count), alarm index (Rate), and alarm index number 3, and the conclusion of the rule is defined as Part and Status, which are phenomena of the case.

As another example, I can see in rule 2 that the conclusion of the case is based on the alarm message of alarm #1 and the conclusion of the rule is linked to the knowledge of the process map.

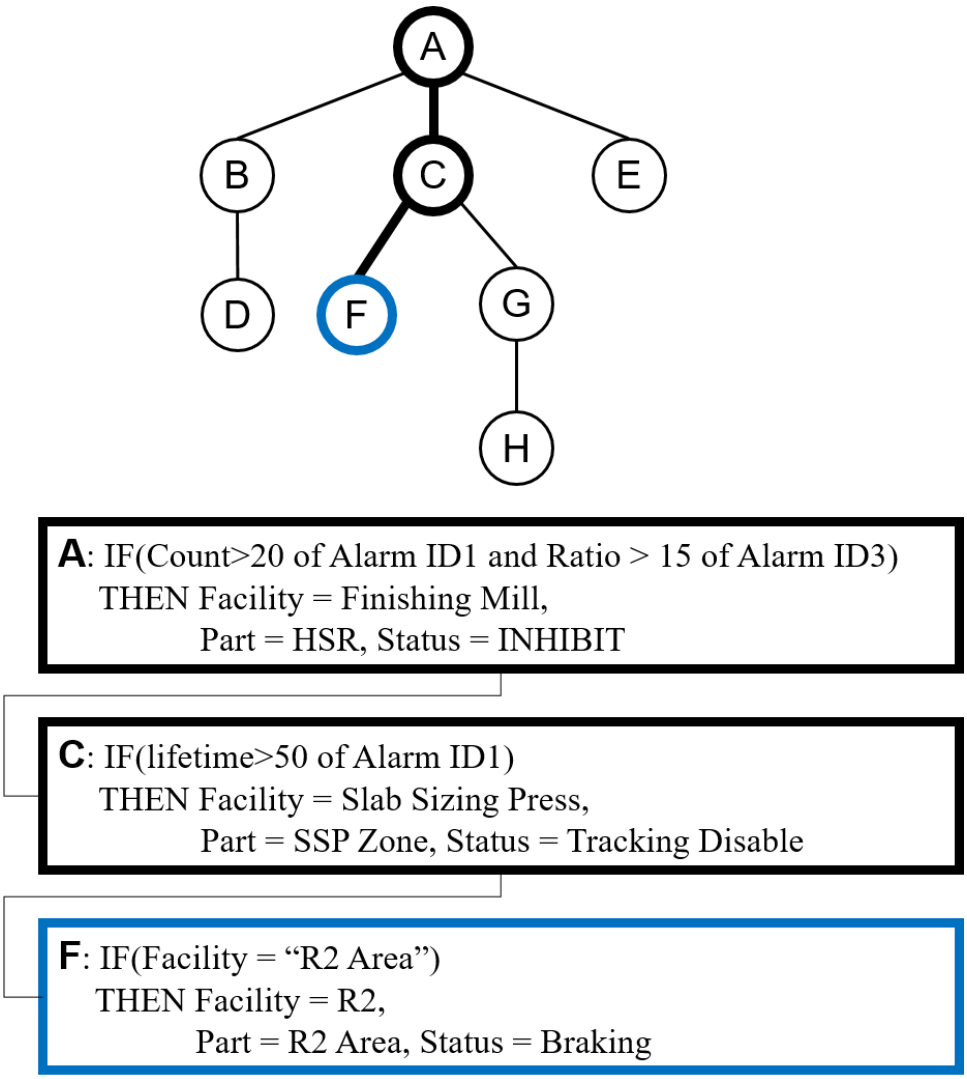


Figure 6.5 The structure of Knowledge Base

6.3 Preventive Management System

The proposing preventive maintenance system monitors the current state of the facility with real-time alarms. It is also capable of conducting failure diagnosis and failure prediction via historic failure cases which is in causal-relationship format. Figure 6.6 represents an example of the operation of the preventive maintenance system. Four alarms are included in the alarm list collected during an hour, and the alarm list is used as the input case for the failure prediction system. Since the rule for the first alarm is stored in the knowledge base, inference engine evaluates the alarm list with the rule corresponding to the first alarm from the knowledge base and obtains the result when the rule is satisfied. The result of the rule is a failure phenomenon, and it represents that a certain phenomenon will occur at the target facility SSP zone. In order to predict possible failures from the acquired failure phenomenon, the system searches for failure cases which contains an equivalent failure phenomenon from the process map. The system provides the failure cases related with the facility Slab Sizing Press Area, the designated facility in the alarm, to the field expert who is monitoring the alarm. Field expert checks the phenomenon of the problematic facility and actions they can take. They are also capable of analyzing the cause of the failure by tracking the previous failure case from backward.

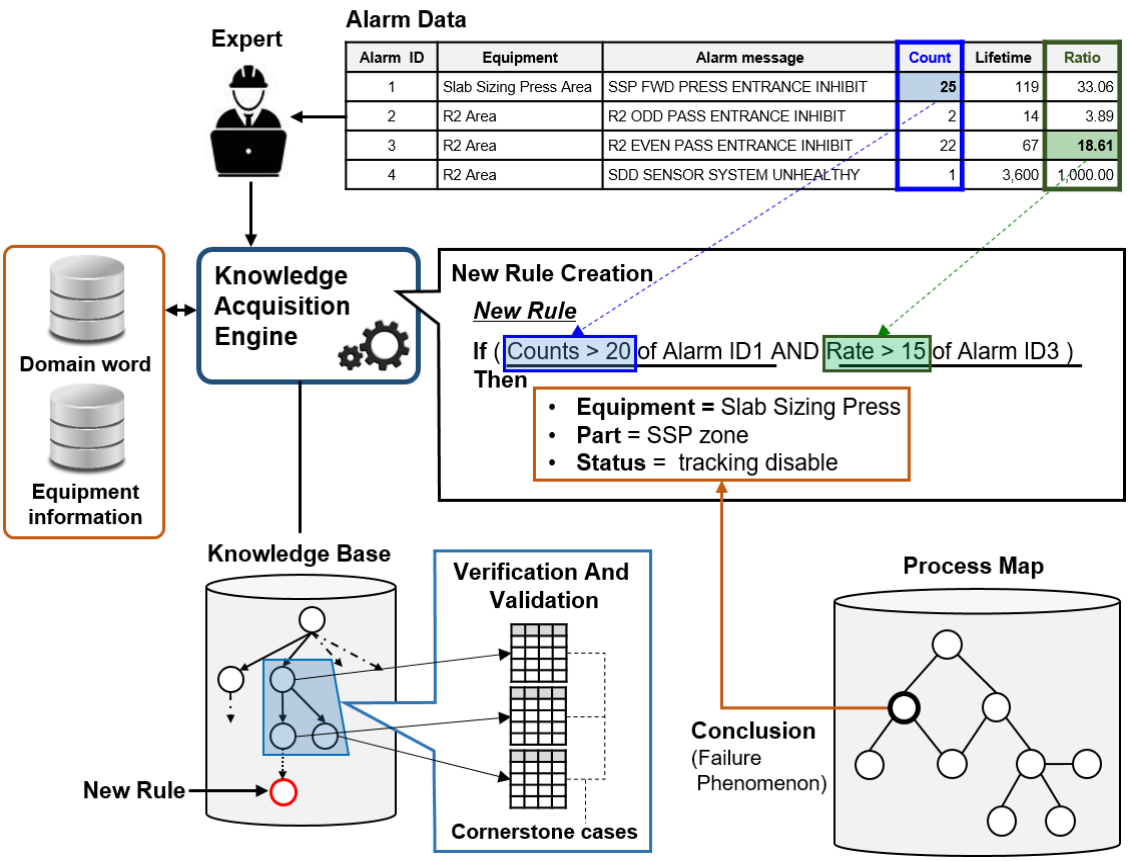


Figure 6.6 A conceptual diagram of knowledge acquisition and representation process for the proposed framework

6.4 Implementation

In this thesis, the example process of the proposed system is described, which currently operates in the steelworks industry. Figure 6.7 shows the implemented screen of the preventive maintenance system. In the alarm occurrence area located on the upper part of the screen, the occurrence status of the alarm is displayed every hour. It also displays the number of predictions for failures when the alarm occurred in the alarm occurrence area, which obtained using alarm data and failure knowledge. Users can check the details of the alarm occurrence and failure cases for the alarm via the user interface by adjusting the specific time zone. In the left bottom side of the screen, details of the alarm occurrence are displayed, including the start time and end time of the alarm, a facility which signaled the alarm, contents of the alarm, and the ratio that the individual alarm possesses. The alarms are classified as facility alarm or operation alarm. In the right bottom side of the screen, failure occurrence detail panel shows the details of the failures which occurred in the time zone that the user selected. The failure prediction result is the inference result about the alarm that shows the SSP entrance prohibited status. The failure related alarm table shows the alarm which is primarily used during the inference process. The pseudo-failure case shows all the failure cases, including the inference result, and it also measures, sorts, and displays the degree of similarity in order to show if the inference result is exactly included. Figure 6.8 represents the details of the failure case, which is displayed when the user selects a specific failure case from a failure case list in order to add a new knowledge. On the left side, the contents of the failure cases, which are generated by analyzing failure reports, sorted in the causal relationship. Each failure phenomenon is classified as failure and action, based on its attributes, which in this example, enables field experts to predict when SSP, SSP PRE-FORMING 불능 will occur (Figure 6.8). Based on those steps, the field experts can check the problem and add new failure prediction knowledge.

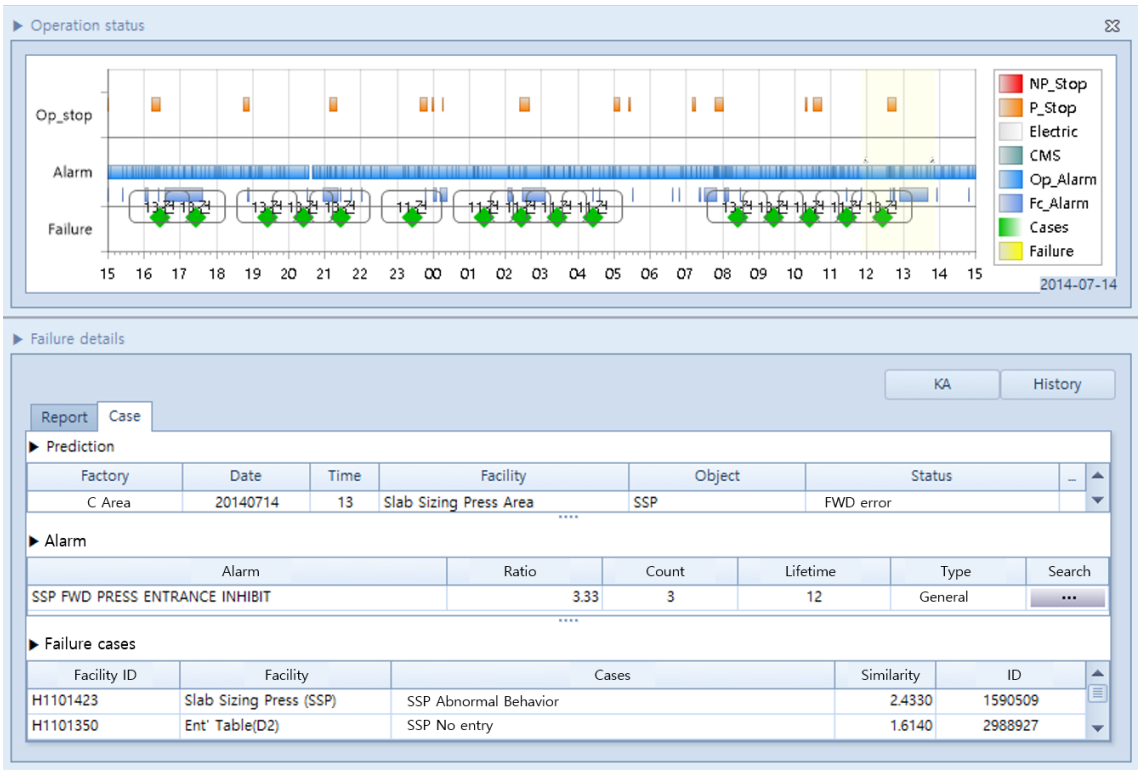


Figure 6.7 The failure prediction and prevention interface

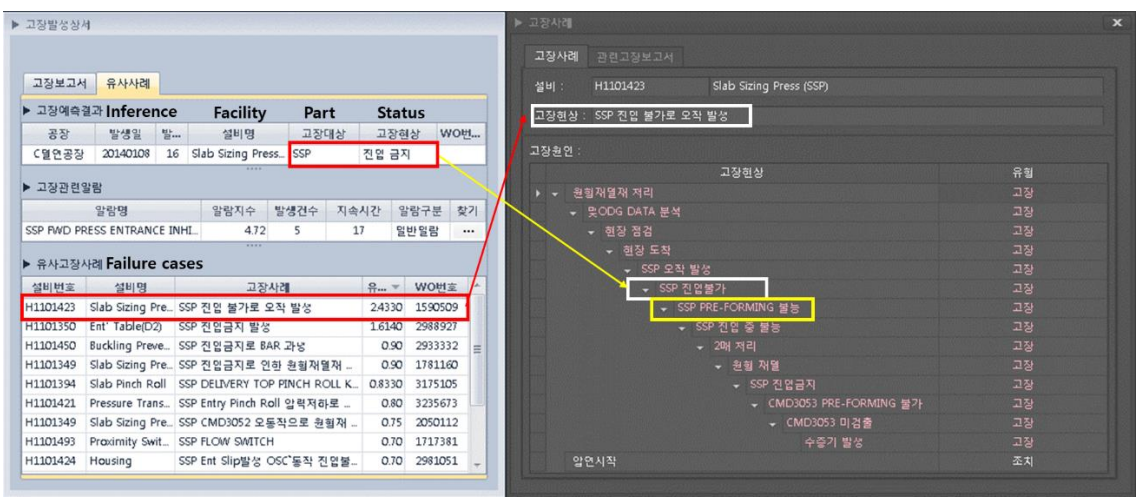


Figure 6.8 The failure case detail view for a specific alarm

Algorithm 7 : Mapping inference result with process map

```
1:  LET String range ← similar phenomena search range
2:  LET String inf_grp_id ← inference group id
3:  LET Array entry ← a range of data related to inferred phenomena
4:  LET String eqp_num ← equipment id of the entry
5:  LET Array initial_similar_phe ← the group of similar phenomenon based on
6:                                  equipment number
7:  LET INT sim_rate_threshold ← the similarity rate threshold (0.5)
8:  LET Array final_similar_phe ← filtered similar phenomenon based on the similarity
9:                                  rate threshold

10: FUNCTION getPhenomenon(range, inf_grp_id)
11: entry ← Query getEntry(inf_grp_id)
12: eqp_num ← getEquipmentNum(entry)
    initial_similar_phe ← Query getSimilarphe(eqp_num, range)
13: for all phenomenon phe in initial_similar_phe do
14:     similarity rate sim ← calculateSimilarity(phe)
15:     if sim ≥ sim_rate_threshold then
16:         push(final_similar_phe, phe)
17:     end if
18: end for final_similar_phe
```

Algorithm 7 explains the procedure for failure detection. I am creating a string variable range which holds the similar phenomena search range. This variable keeps the search specific and efficient by clearly pointing out the data on which the search is run on. Then another string variable inf_group_id is created to hold the value of inference group id. This group id is later used to perform search on specific inference groups. Then an array variable entry is created which holds a range of data related to inference phenomena. All the data to be put in this variable will come from the entries of the specified inference group ids. Each entry is associated with an equipment which being damaged or which malfunctioning may have caused the failure. The ids of these equipment are stored in a variable named eqp_num. To store the incidents associated with the specified equipment which is also within our range as a result of being similar to our case, I use an array variable named initial similar phenomenon. Then I have an integer named similarity_rate_threshold which is fixed at 0.5. I have one more variable called final_simillar_phe

where the similar phenomenon that satisfies the threshold is stored. A function to get the phenomenon is used which takes range and inference group id as parameters to find out the similar phenomenon from the range. Entries, equipment numbers and initial similar phenomenon are extracted by using getter functions. Then the similarity is calculated for all the variables in the initial similar phenomenon array and compared with the similarity rate threshold. The ones who are greater or equal to the similarity rate threshold are pushed on the stack of final similar phenomenon. Thus, failure detection is complete with the separation of final similar phenomenon.

6.5 Evaluation

The failure prediction performance of the proposed system is evaluated through experiments, in order to prove that the failure cases of the process map indeed include causal relationships.

6.5.1 Experiment data

The data used in the experiment includes the failure reports for the failures that occurred more than once in domestic steelwork, and the alarm data collected 1 hour before and after the failure occurrence. All the alarm data are collected in real-time. In the proposed system, the data is pre-processed in the unit of 1 hour with the number of occurrences, occurred periodically, and the ratio of occurrence. From October 2012 to July 2016, a total number of 502,308 alarm data were collected. The failure report uses the failure cases built in the process map by analyzing 400 failure reports among 713, which includes the failures that occurred more than once, and are collected during the same period. The number of occurrences for the identical failures is not constant, but 4 iterated failures have occurred on average. The knowledge base of the expert system consists of 237 rules built by two field experts. For the experiment, training data and test data were used in a 6 to 4 ratio. 100 failure data and 200,923 alarm data were used as the test data.

Input cases of the expert system are the alarm lists consisting of multiple alarms and which are used for failure prediction and knowledge acquisition. The alarm system collects alarm every hour in real-time and forwards them to the expert system after processing those alarms.

As shown in Table 6.1, the alarm data consist of seven attributes such as facility ID, facility name, alarm ID, alarm name, counts of alarm, the lifetime of the alarm and rate of the alarm.

Table 6.1 The sample testing data: first 10 rows

Alarm ID	Time	Facility ID	Count	Lifetime	Ratio	Status
DRV 183	17	H1103364	1	3228	896.67	INTRUDE
ES 041	16	H1101349	10	112	31.11	HUNTING
MCC 323	23	H1103364	1	3600	1000	IMPACT
APC 014	8	H1101349	4	22	6.11	BUR
PAG 004	1	H1101613	13	43	11.94	LEAK
PRC 090	9	H1101349	4	21	5.83	CARBONIZATION
PRC 058	7	H1105709	1	30	8.33	NORMAL
PRC 071	22	H1102579	1	82	22.78	NO LINK
GRS 008	10	H1105709	1	20	5.56	NO REVERSE
PRC 020	7	H1101613	1	4	1.11	CUT
...	...	...	...	...	...	...

6.5.2 Experiment method

The experiment done here consists of the following two parts. The first part shows the possibility of failure prediction method which is based on the alarm and failure knowledge, by evaluating the success rate of failure prediction with the degree where the inference results of the alarm and failure phenomenon of the failure case are mapped. The second part shows the superiority of the proposed system by comparing failure prediction accuracy of the proposed system and three previous types of research on failure prediction. Failure prediction is performed by inputting alarms into the system in the order of occurrence time, and by evaluating if the order of the reasoning results is equivalent to the order of failure phenomenon in that failure case. The following processes do the evaluation. For the reliability of the constructed alarm knowledge and failure cases, two field experts who are in charge of the facility monitoring at the actual steelworks evaluated the inference result of the alarm generated by the expert system. They compared the order of inference results and failure case to confirm if the cause of the actual failure and occurred failure is equivalent.

6.5.3 Experiment Result

6.5.3.1 Knowledge Analysis

The ten most-frequently satisfied rules are ranked and shown in Table 6.2, highlighting the most frequent failure cases. The most-frequently rule was the normal (rule 0) which does not have any failure to predict. Among 400 failure type, the system found LEAK (rule 17 - If the count is larger than 9 and lifetime is more than 8 hours, Then the failure is LEAK) and BURR (rule 201 - If lifetime is less than 2 hours, Then the failure is BURR).

Table 6.2 Top 10 satisfied rules

No.	Frequency	Rule ID	Failure Description
1	13.87%	0	Default (root) rule
2	9.98%	2	Detect the LEAK using the starting time
3	8.87%	17	Detect the BURN using lifetime
4	3.48%	201	Detect the DEFECTION using alarm id
5	2.99%	38	Detect the HUNTING using lifetime and facility id
6	2.61%	120	Detect the NO LINK using ratio and count
7	2.02%	7	Detect the NO REVERSE using alarm id and count
8	1.81%	79	Detect the SLIP using lifetime and ratio
9	1.47%	22	Detect the TRANSFORM using lifetime
10	1.03%	19	Detect the GAP using alarm id and facility id
...	...	...	...

Figure 6.9 shows the seasonal frequency of satisfied rules coupled with the real depth (which is their level in the decision tree), indicating failure prediction conceptual depth. Note that the root (default) rule is level 1. As can be seen in the figure, the most common rule depth is 4. Rules at this depth level includes the combination of various types of attribute sets.

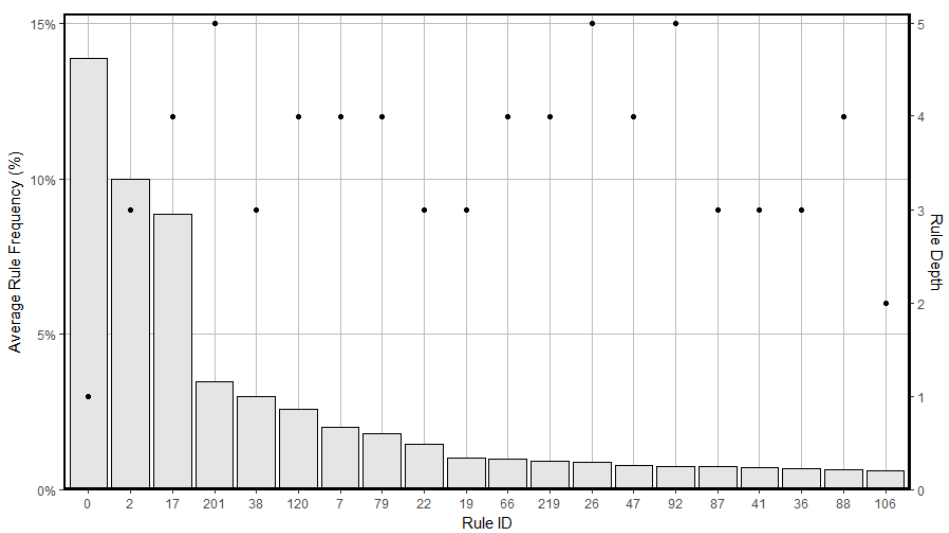


Figure 6.9 Inferred Rule Frequency and Depth

Figure 6.10 shows the rate at which a rule is fired when a fault is detected in the knowledge base. 13.87% of the alarms were fired in Rule 0, which is the default rule with no knowledge of faults. On the other hand, 86.13% were fired among rule 1 ~ 236 related to the failure. This rate was the same as normal, not faulty.

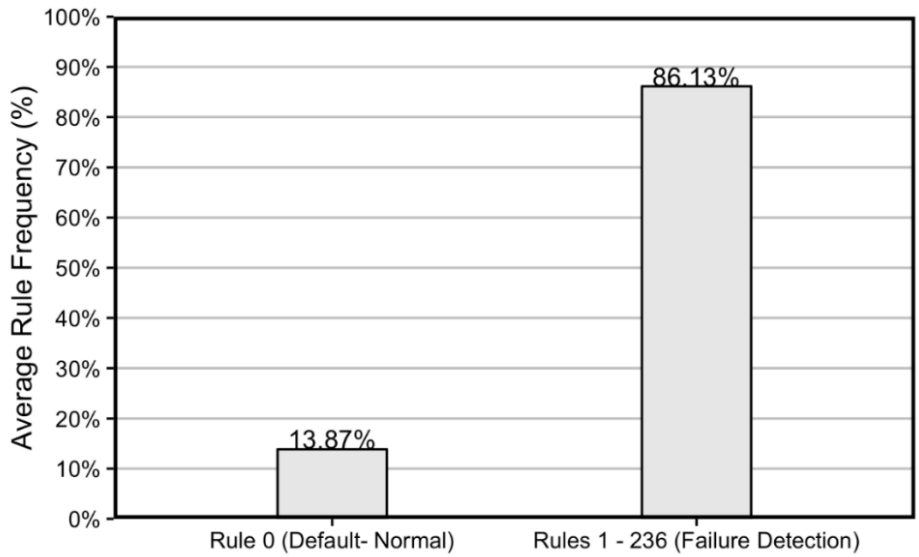


Figure 6.10 Ratio of Default and Failure Rules

6.5.3.2 Performance evaluation of failure prediction

To evaluate the superiority of the proposing failure prediction method, I conducted a comparison with the previous failures prediction methods. The studies compared with the paper are as follows:

- Santos et al. (2010) [148]: The authors proposed prediction method based on machine learning, in order to predict the major failures in a casting factory domain. They compared Bayesian network, SVM, and decision tree, and concluded that decision tree has the high prediction success rate.

- Liu and Jiang (2008) [149]: The authors used particle filter, which is frequently applied in signal processing, Bayesian inference, and machine learning. The research tried to predict failures with a hybrid system in the discrete-continuous composite environment.

- Chen et al. (2015) [150]: The authors used knowledge-based neural fuzzy inference for the failure prediction of turbines in wind power plant. The performance of failure prediction method proposed in this thesis is evaluated by comparing with the algorithms of [148], [149], and [150] in an equivalent environment. To be specific, decision tree of [148], particle filter of [149], and neural fuzzy inference of [150] are used with the alarm data proposed in this thesis. On Table 6.3, failure prediction accuracy between each experiment are compared.

Table 6.3 Review of Failure Prediction By Previous Failure Prediction System

Author	Description	Accuracy
Santos et al. (2010)	Applied different machine learning techniques (incl. Bayesian Network, SVM, and decision tree)	81.4%
Liu and Jiang (2008)	Used particle filter with Bayesian Inference	64.2%
Chen et al. (2015)	Applied knowledge-based neural fuzzy inference	90.3%
Proposed System	Natural Language-based Processing Map + knowledge-based alarm prediction system	**95.7%**

Due to the nature of alarm data, the decision tree of [148] classified various variables into exact horizontal and vertical relations, which is not adequate for classification. As a result, the complexity of the tree grows rapidly and pruning becomes hard, which resulted in an accuracy of 81.4%, slightly lower than the result of applying general decision tree. Since the particle filter used in [149] is based on Monte-Carlo method, which requires a massive amount of sample data, [149] showed the lowest accuracy of 64.2%. The neural-fuzzy inference method of [150] is a method which introduced learning ability of neural network into the conventional fuzzy logic method. Since it is a method which enables continuous learning by granting the learning ability to the expert knowledge-based fuzzy logic system, its key features are well utilized in the processing of complex and continuously accumulated real-time alarm data, which is reflected when it shows the similarities result recorded in its paper, 90.3%. The failure prediction success rate of the proposed method is 95.7%, which is superior to the methods used in the comparison. The reason behind this is that firstly the experts who have diverse experience in failures constructed the knowledge base directly, and secondly, the knowledge which contains real failure cases and a causal relationship is used to predict failures. Therefore, I can interpret the high accuracy as a result of utilization of high-quality knowledge which appropriately represents the actual failure cases.

6.6 Conclusion

The preventive maintenance system proposed in this thesis is an effective alternative directly related to the popular smart factory for two reasons. It is based on the knowledge of experts, which greatly lowers the dependency towards human labor, and it enables effective failure prediction and diagnosis by the system. The proposed system can be utilized in various domains since it focused on the knowledge of experts which was not easily reusable before in specific domains. Proposed failure analysis system is meaningful in a perspective that it suggests new methods for knowledge sharing and generalization, which was long considered impossible. The knowledge of facilities or failures are easily obtained from manuals or reports, but the problem was that the total amount of information was enormous, and it was not easy to find exactly the information I wanted. With those reasons, the usability of failure reports went down, which made many field experts write the failure reports perfunctorily, resulting in lower low quality of reports. Such problems can be solved by improving the working environment with the help of the system. Although the technical approach of the system is meaningful to a certain extent, the system must be understood in order to be utilized in actual operation. The problem with the current failure report is that it is not easy to understand the reports for both human and systems, due to the excessive use of shortened words and specific terminology. If the human understands the working process of the system, they will write the failure reports in a way that the system could understand, which can result in a fairly accurate analysis of the system, which will reduce the dependency towards human labor. If the quality and usability of failure cases go up, the usefulness of expert system which interoperates with failure cases will also go up. In order to utilize the failure cases, the field experts will accumulate the experiential knowledge of alarm into the knowledge base regularly, and if the knowledge with high accuracy is continuously gathered to a certain extent, the system could utilize such expert knowledge to improve the accuracy of failure prediction and diagnosis with alarms. Therefore, with this process, I can overcome the disasters and human injuries, by maximizing the efficiency of preventive maintenance in the actual industrial field.

7 Study Conclusion and Future Directions

7.1 Summary and Conclusion

In this dissertation, I studied how to use alarm data and expert knowledge together with these characteristics. In this research, build knowledge by using the failure report reflects the significant knowledge resource, alarm data, expert knowledge, and expert knowledge of the industry, and proposes a way to continue to manage and use such knowledge.

Firstly, the proposed approach in this thesis allows human experts to incrementally add and maintain the knowledge in the knowledge base without having to rebuild or re-initialise the knowledge base, unlike pure machine learning approaches, which rebuild the knowledge base from scratch each time. Moreover, the proposed failure detection framework can reduce the time of human expertise acquisition and the cost of solving over-generalization and over-fitting problems in machine learning technique. The proposed failure detection framework has never been reported previously. Moreover, this framework can be successful detection approach in the domain if it requires handling big size of the dataset and human expertise.

Secondly, the proposed network-based process map can be utilized in various domains since it focused on the knowledge of experts, which was not easily reusable before in specific domains. Proposed failure analysis system using natural language processing is meaningful in a perspective that it suggests new methods for knowledge sharing and generalization, which was long considered impossible.

Finally, by fusing the knowledge and causal relation for failure, hybrid knowledge engineering methods were applied to failure diagnosis and prediction. As a result of the performance analysis, the proposed framework is superior to the other methodologies regarding failure diagnosis and prediction.

If the quality and usability of failure cases go up, the usefulness of expert system which interoperates with failure cases will also go up. In order to utilize the failure cases, the field experts will accumulate the experiential knowledge of alarm into the knowledge base regularly, and if the knowledge with high accuracy is continuously gathered to a certain extent, the system could utilize such expert knowledge to improve the accuracy of failure prediction and diagnosis with alarms. Therefore, with this process, I can overcome the disasters and human injuries, by maximizing the efficiency of preventive maintenance in the actual industrial field.

7.2 Future Work

First of all, the proposed framework in this research has been evaluated by the industrial data collected from Hyundai Co, and the density of dataset is still limited. In the future, the proposed approach will be investigated with larger dataset in not only industrial domain. Expanding the experiment with large dataset in different domain, i.e. network alarm domain, would discover the performance and potentiality of the proposed framework as an alternative solution of existing machine learning-based or human-based knowledge base modelling approaches.

Secondly, the proposed framework was produced the failure prediction result by using the process map, which includes the cause-and-effect procedural knowledge. Process map follows the concept of a network-based knowledge representation. The proposed approach applied various similarity measure algorithm in order to extract the instances from human-written failure report to this network-based knowledge base. For the future work, I will review the trend of instance matching algorithm, evaluate it with the current domain, and find better algorithm than simple similarity measure techniques.

Finally, the thesis has proposed the novel solution of knowledge learning and maintenance in the industrial domain, especially large industrial plant management so the following industrial engineering tasks will be conducted in the future.

- Improve the quality of knowledge by using FMEA report and Fishbone diagram report, which are valuable in terms of information accuracy and fidelity
- Calculate the probability of rule acceptance by applying Average Run Length (ARL) or Average Time to Signal (ATS) in order to evaluate the accuracy of knowledge.
- Evaluate the performance by using out-of-control performance and in-control performance, and analyzing its trade-off.
- Develop the semi-automatic human rule creation approach by applying the stochastic aspects of failures
- Improve the performance by using exemplar-based model or prototype-based model.

Bibliography

[1] Langone, R., Alzate, C., Bey-Temsamani, A., & Suykens, J. A. (2014, December). Alarm prediction in industrial machines using autoregressive LS-SVM models. In Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on (pp. 359-364). IEEE.

[2] Foong, O. M., Sulaiman, S. B., Rambli, D. R. B. A., & Abdullah, N. S. B. (2009, October). ALAP: Alarm prioritization system for oil refinery. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 2, pp. 2-7). San Francisco.

[3] Cheng, Y., Izadi, I., & Chen, T. (2013). Optimal alarm signal processing: Filter design and performance analysis. IEEE Transactions on Automation Science and Engineering, 10(2), 446-451.

[4] Pham, H. N. A., & Triantaphyllou, E. (2008). The impact of overfitting and overgeneralization on the classification accuracy in data mining. In Soft Computing for Knowledge Discovery and Data Mining (pp. 391-431). Springer US.

[5] Compton, P., & Jansen, R. (1990). Knowledge in context: A strategy for expert system maintenance. AI'88, 292-306.

[6] Zeltmate, I., & Grundspenkis, J. (2010). An extension of frame-based knowledge representation schema. In Proc. of the Int. Multi-Conf. on Complexity, Informatics and Cybernetics (Vol. 1, pp. 6-9).

[7] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185.

[8] Kangho Roh, Jin Wook Kim, Eunsang Kim, Kunsoo Park, Hwan-Gue Cho (2010). Edit Distance Problem for the Korean Alphabet. Journal of KIISE : Computer Systems and Theory 37(2), 2010.4, 103-109

[9] Kangho Roh, Kunsoo Park, Hwan-Gue Cho, Sowon Chang (2011). Similarity and Edit Distance Algorithms for the Korean Alphabet using One-Dimensional Array of Phonemes. Journal of KIISE : Computing Practices and Letters 17(10), 2011.10, 519-526

[10] Zhou, G., & Su, J. (2002). Named entity recognition using an HMM-based chunk tagger.

Paper presented at the proceedings of the 40th Annual Meeting on Association for Computational Linguistics.

[11] Skounakis, M., Craven, M., & Ray, S. (2003). Hierarchical hidden Markov models for information extraction. Paper presented at the IJCAI.

[12] GuoDong, Z., & Jian, S. (2004). Exploring deep knowledge resources in biomedical name recognition. Paper presented at the Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications.

[13] Ponomareva, N., Pla, F., Molina, A., & Rosso, P. (2007). Biomedical named entity recognition: a poor knowledge HMM-based approach Natural Language Processing and Information Systems (pp. 382-387): Springer.

[14] Todorovic, B. T., Rancic, S. R., Markovic, I. M., Mulalic, E. H., & Ilic, V. M. (2008). Named entity recognition and classification using context Hidden Markov Model. Paper

[15] Jin, W., Ho, H. H., & Srihari, R. K. (2009). A novel lexicalized HMM-based learning framework for web opinion mining. Paper presented at the Proceedings of the 26th annual international conference on machine learning.

[16] Reynar, J. C., & Ratnaparkhi, A. (1997). A maximum entropy approach to identifying sentence boundaries. Paper presented at the Proceedings of the fifth conference on Applied natural language processing.

[17] Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. Paper presented at the Proceedings of the ACL 2004 on Interactive poster and demonstration sessions.

[18] Xinhao, W., Xiaojun, L., Dianhai, Y., Hao, T., & Xihong, W. (2006). Chinese word segmentation with maximum entropy and n-gram language model. Paper presented at the COLING• ACL 2006.

[19] Benajiba, Y., Rosso, P., & Benedíruiz, J. (2007). Anersys: An arabic named entity recognition system based on maximum entropy. Computational Linguistics and Intelligent Text Processing, 143-153.

[20] Konkol, M., & Konopík, M. (2011). Maximum entropy named entity recognition for czech language. Paper presented at the Text, Speech and Dialogue.

[21] Ahmed, I., & Sathyaraj, R. (2015). Named entity recognition by using maximum

entropy. International Journal of Database Theory and Application, 8(2), 43-50.

[22] Yi, E., Lee, G. G., Song, Y., & Park, S.-J. (2004). SVM-Based Biological Named Entity Recognition Using Minimum Edit-Distance Feature Boosted by Virtual Examples. Paper presented at the IJCNLP.

[23] Li, Y., Bontcheva, K., & Cunningham, H. (2005). SVM based learning system for information extraction. Lecture notes in computer science, 3635, 319-340.

[24] Aramaki, E., Imai, T., Miyo, K., & Ohe, K. UTH: Semantic Relation Classification using Physical Sizes.

[25] Benajiba, Y., Diab, M., & Rosso, P. (2008). Arabic named entity recognition: An svm-based approach. Paper presented at the Proceedings of 2008 Arab International Conference on Information Technology (ACIT).

[26] Habib, M. S., & Kalita, J. (2010). Scalable biomedical Named Entity Recognition: investigation of a database-supported SVM approach. International journal of bioinformatics research and applications, 6(2), 191-208.

[27] Li, D., Kipper-Schuler, K., & Savova, G. (2008). Conditional random fields and support vector machines for disorder named entity recognition in clinical texts. Paper presented at the Proceedings of the workshop on current trends in biomedical natural language processing.

[28] Singh, T. D., Nongmeikapam, K., Ekbal, A., & Bandyopadhyay, S. (2009). Named Entity Recognition for Manipuri Using Support Vector Machine. Paper presented at the PACLIC.

[29] Cai, P., Luo, H., & Zhou, A. (2010). Semantic Entity Detection by Integrating CRF and SVM. Paper presented at the WAIM.

[30] Habib, M. S. (2008). Addressing scalability issues of named entity recognition using Multi-class Support Vector Machines. World Academy of Science, Engineering and Technology.–2008.–37.–P, 69-78.

[31] Ju, Z., Wang, J., & Zhu, F. (2011). Named entity recognition from biomedical text using SVM. Paper presented at the Bioinformatics and Biomedical Engineering,(iCBBE) 2011 5th International Conference on.

[32] Björne, J., Kaewphan, S., & Salakoski, T. (2013, June). UTurku: Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge. In SemEval@ NAACL-HLT (pp. 651-659).

[33] Paliouras, G., Karkaletsis, V., Petasis, G., & Spyropoulos, C. D. (2000). Learning decision trees for named-entity recognition and classification. Paper presented at the ECAI Workshop on Machine Learning for Information Extraction.

[34] Isozaki, H. (2001). Japanese named entity recognition based on a simple rule generator and decision tree learning. Paper presented at the Proceedings of the 39th Annual Meeting on Association for Computational Linguistics.

[35] Black, W. J., & Vasilakopoulos, A. (2002). Language independent named entity classification by modified transformation-based learning and by decision tree induction. Paper presented at the proceedings of the 6th conference on Natural language learning-Volume 20.

[36] Witschel, H. F. (2005). Using decision trees and text mining techniques for extending taxonomies. Paper presented at the Learning and Extending Lexical Ontologies by using Machine Learning Methods, Workshop at ICML-05.

[37] Szarvas, G., Farkas, R., & Kocsor, A. (2006). A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms. Paper presented at the International Conference on Discovery Science.

[38] Zhou, X., Han, H., Chankai, I., Prestrud, A., & Brooks, A. (2006). Approaches to text mining for clinical medical records. Paper presented at the Proceedings of the 2006 ACM symposium on Applied computing.

[39] Chakaravarthy, V. T., Pandit, V., Roy, S., Awasthi, P., & Mohania, M. (2007). Decision trees for entity identification: Approximation algorithms and hardness results. Paper presented at the Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.

[40] Abdallah, S., Shaalan, K., & Shoaib, M. (2012). Integrating rule-based system with classification for arabic named entity recognition. Computational Linguistics and Intelligent Text Processing, 311-322.

[41] Oudah, M., & Shaalan, K. F. (2012). A Pipeline Arabic Named Entity Recognition using a Hybrid Approach. Paper presented at the Coling.

[42] Prokofyev, R., Demartini, G., & Cudré-Mauroux, P. (2014). Effective named entity recognition for idiosyncratic web collections. Paper presented at the Proceedings of the 23rd

international conference on World wide web.

[43] McDonald, R., & Pereira, F. (2005). Identifying gene and protein mentions in text using conditional random fields. BMC bioinformatics, 6(1), S6.

[44] Okanohara, D., Miyao, Y., Tsuruoka, Y., & Tsujii, J. i. (2006). Improving the scalability of semi-markov conditional random fields for named entity recognition. Paper presented at the Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.

[45] Peng, F., & McCallum, A. Accurate information extraction from research papers using conditional random fields. Retrieved on April 13, 2013.

[46] Zhao, H., Huang, C., Li, M., & Lu, B.-L. (2006). Effective Tag Set Selection in Chinese Word Segmentation via Conditional Random Field Modeling. Paper presented at the PACLIC.

[47] Bundschus, M., Dejori, M., Stetter, M., Tresp, V., & Kriegel, H.-P. (2008). Extraction of semantic biomedical relations from text using conditional random fields. BMC bioinformatics, 9(1), 207.

[48] Sobhana, N., Mitra, P., & Ghosh, S. (2010). Conditional random field based named entity recognition in geological text. International Journal of Computer Applications, 1(3), 143-147.

[49] Rocktäschel, T., Weidlich, M., & Leser, U. (2012). ChemSpot: a hybrid system for chemical named entity recognition. Bioinformatics, 28(12), 1633-1640.

[50] Li, Y., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., & Jagadish, H. (2008). Regular expression learning for information extraction. Paper presented at the Proceedings of the Conference on Empirical Methods in Natural Language Processing.

[51] Brauer, F., Rieger, R., Mocan, A., & Barczynski, W. M. (2011). Enabling information extraction by inference of regular expressions from sample entities. Paper presented at the Proceedings of the 20th ACM international conference on Information and knowledge management.

[52] Ek, T., Kirkegaard, C., Jonsson, H., & Nugues, P. (2011). Named entity recognition for short text messages. Procedia-Social and Behavioral Sciences, 27, 178-187.

[53] He, G., Zhang, Y., & Wu, X. (2013). Information extraction of forum based on regular

expression. Paper presented at the Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on.

[54] Sawsaa, A., & Lu, J. (2011). Extracting information science concepts based on jape regular expression. Paper presented at the In: WORLDCOMP'11The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing.

[55] Tanabe, L., Xie, N., Thom, L. H., Matten, W., & Wilbur, W. J. (2005). GENETAG: a tagged corpus for gene/protein named entity recognition. BMC bioinformatics, 6(1), S3.

[56] Szarvas, G., Farkas, R., Felföldi, L., Kocsor, A., & Csirik, J. (2006). A highly accurate Named Entity corpus for Hungarian. Paper presented at the Proceedings of International Conference on Language Resources and Evaluation.

[57] Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., & Salakoski, T. (2007). BioInfer: a corpus for information extraction in the biomedical domain. BMC bioinformatics, 8(1), 50.

[58] Rosenfeld, B., & Feldman, R. (2007). Using corpus statistics on entities to improve semi-supervised relation extraction from the web. Paper presented at the ACL.

[59] Tomanek, K., Wermter, J., & Hahn, U. (2007). An Approach to Text Corpus Construction which Cuts Annotation Costs and Maintains Reusability of Annotated Data. Paper presented at the EMNLP-CoNLL.

[60] Ekbal, A., & Bandyopadhyay, S. (2008). A web-based Bengali news corpus for named entity recognition. Language Resources and Evaluation, 42(2), 173-182.

[61] Kipper-Schuler, K., Kaggal, V., Masanz, J., Ogren, P., & Savova, G. (2008). System evaluation on a named entity corpus from clinical notes. Paper presented at the Language resources and evaluation conference, LREC.

[62] Roberts, A., Gaizauskas, R., Hepple, M., Demetriou, G., Guo, Y., Setzer, A., & Roberts, I. (2008). Semantic annotation of clinical text: The CLEF corpus. Paper presented at the Proceedings of the LREC 2008 workshop on building and evaluating resources for biomedical text mining.

[63] Ohta, T., Kim, J.-D., Pyysalo, S., Wang, Y., & Tsujii, J. i. (2009). Incorporating GENETAG-style annotation to GENIA corpus. Paper presented at the Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing.

[64] Desmet, B., & Hoste, V. (2010). Towards a balanced named entity corpus for dutch. Paper presented at the 7th Conference on International Language Resources and Evaluation (LREC 2010).

[65] Lee, S., & Lee, G. G. (2004). A Bootstrapping Approach for Geographic Named Entity Annotation. Paper presented at the AIRS.

[66] Kozareva, Z. (2006). Bootstrapping named entity recognition with automatically generated gazetteer lists. Paper presented at the Proceedings of the eleventh conference of the European chapter of the association for computational linguistics: student research workshop.

[67] Pennacchiotti, M., & Pantel, P. (2006). A bootstrapping algorithm for automatically harvesting semantic relations. Proceedings of Inference in Computational Semantics (ICoS-06), 87-96.

[68] Van Erp, M. (2006). Bootstrapping multilingual geographical gazetteers from corpora. Paper presented at the ESSLLI Student Session.

[69] Arguello, J., & Callan, J. (2007). A bootstrapping approach for identifying stakeholders in public-comment corpora. Paper presented at the Proceedings of the 8th annual international conference on Digital government research: bridging disciplines & domains.

[70] Lee, S., & Lee, G. G. (2007). Exploring phrasal context and error correction heuristics in bootstrapping for geographic named entity annotation. Information Systems, 32(4), 575-592.

[71] Dang, V. B., & Aizawa, A. (2008). Multi-class named entity recognition via bootstrapping with dependency tree-based patterns. Paper presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining.

[72] Kawai, H., Mizuguchi, H., & Tsuchida, M. (2008). Cost-effective web search in bootstrapping for named entity recognition. Paper presented at the International Conference on Database Systems for Advanced Applications.

[73] Venturi, G., Montemagni, S., Marchi, S., Sasaki, Y., Thompson, P., McNaught, J., & Ananiadou, S. (2009). Bootstrapping a verb lexicon for biomedical information extraction. Paper presented at the International Conference on Intelligent Text Processing and Computational Linguistics.

[74] Wu, D., Lee, W. S., Ye, N., & Chieu, H. L. (2009). Domain adaptive bootstrapping for named entity recognition. Paper presented at the Proceedings of the 2009 Conference on Empirical

Methods in Natural Language Processing: Volume 3-Volume 3.

[75] Polifroni, J., Kiss, I., & Adler, M. (2010). Bootstrapping Named Entity Extraction for the Creation of Mobile Services. Paper presented at the LREC.

[76] Glass, M., & Barker, K. (2011). Bootstrapping relation extraction using parallel news articles. Paper presented at the Proceedings of the IJCAI Workshop on Learning by Reading and its Applications in Intelligent Question-answering, Barcelona.

[77] Putthividhya, D. P., & Hu, J. (2011). Bootstrapped named entity recognition for product attribute extraction. Paper presented at the Proceedings of the Conference on Empirical Methods in Natural Language Processing.

[78] Schone, P., Allison, T., Giannella, C., & Pfeifer, C. (2011). Bootstrapping multilingual relation discovery using english wikipedia and wikimedia-induced entity extraction. Paper presented at the Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on.

[79] Sun, A., & Grishman, R. (2011). Cross-domain bootstrapping for named entity recognition. Balog et al.[3], 33-40.

[80] Teixeira, J., Sarmento, L., & Oliveira, E. C. (2011). A Bootstrapping Approach for Training a NER with Conditional Random Fields. Paper presented at the EPIA.

[81] Ashley, C. S., Granatelli, D. B., Cheung, J. C. H., Shepherd, D. M., Weiss, R. A., & Schultz, B. I. (2016). U.S. Patent No. 9,355,477. Washington, DC: U.S. Patent and Trademark Office.

[82] Nan, C., Khan, F., & Iqbal, M. T. (2008). Real-time fault diagnosis using knowledge-based expert system. Process safety and environmental protection, 86(1), 55-71.

[83] Abele, L., Anic, M., Gutmann, T., Folmer, J., Kleinsteuber, M., & Vogel-Heuser, B. (2013). Combining knowledge modeling and machine learning for alarm root cause analysis. IFAC Proceedings Volumes, 46(9), 1843-1848.

[84] Aizpurua, O., Galan, R., & Jimenez, A. (2008, April). A new cognitive-based massive alarm management system in electrical power administration. In Devices, Circuits and Systems, 2008. ICCDCS 2008. 7th International Caribbean Conference on (pp. 1-6). IEEE.

[85] Zhao, W., Bai, X., Wang, W., & Ding, J. (2005). A novel alarm processing and fault diagnosis expert system based on BNF rules. In Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES (pp. 1-6). IEEE.

[86] Ebersbach, S., & Peng, Z. (2008). Expert system development for vibration analysis in machine condition monitoring. Expert systems with applications, 34(1), 291-299.

[87] Schleburg, M., Christiansen, L., Thornhill, N. F., & Fay, A. (2013). A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system. Journal of process control, 23(6), 839-851.

[88] Folmer, J., & Vogel-Heuser, B. (2012, March). Computing dependent industrial alarms for alarm flood reduction. In Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on (pp. 1-6). IEEE.

[89] Ahmed, K., Izadi, I., Chen, T., Joe, D., & Burton, T. (2013). Similarity analysis of industrial alarm flood data. IEEE Transactions on Automation Science and Engineering, 10(2), 452-457.

[90] Izadi, I., Shah, S. L., Shook, D. S., Kondaveeti, S. R., & Chen, T. (2009). A framework for optimal design of alarm systems. IFAC Proceedings Volumes, 42(8), 651-656.

[91] Zhu, J., Shu, Y., Zhao, J., & Yang, F. (2014). A dynamic alarm management strategy for chemical process transitions. Journal of Loss Prevention in the Process industries, 30, 207-218.

[92] Yin, G., Zhang, Y. T., Li, Z. N., Ren, G. Q., & Fan, H. B. (2014). Online fault diagnosis method based on incremental support vector data description and extreme learning machine with incremental output structure. Neurocomputing, 128, 224-231.

[93] Wong, P. K., Yang, Z., Vong, C. M., & Zhong, J. (2014). Real-time fault diagnosis for gas turbine generator systems using extreme learning machine. Neurocomputing, 128, 249-257.

[94] Wenyi, L., Zhenfeng, W., Jiguang, H., & Guangfeng, W. (2013). Wind turbine fault diagnosis method based on diagonal spectrum and clustering binary tree SVM. Renewable Energy, 50, 1-6.

[95] Muralidharan, V., Sugumaran, V., & Indira, V. (2014). Fault diagnosis of monoblock centrifugal pump using SVM. Engineering Science and Technology, an International Journal, 17(3), 152-157. [95]

[96] Li, C., & Zhou, J. (2014). Semi-supervised weighted kernel clustering based on gravitational search for fault diagnosis. ISA transactions, 53(5), 1534-1543.

[97] Chivala, D., Mendonça, L. F., Sousa, J. M., & da Costa, J. S. (2010). Application of evolving

fuzzy modeling to fault tolerant control. Evolving Systems, 1(4), 209-223.

[98] Swartout, B., Patil, R., Knight, K., & Russ, T. (1996, November). Toward distributed use of large-scale ontologies. In Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems (pp. 138-148).

[99] Fernández-López, M. (1999). Overview of methodologies for building ontologies, Retrieved from: http://oa.upm.es/5480/1/Overview_Of_Methodologies.pdf

[100] Uschold, M., & King, M. (1995). Towards a methodology for building ontologies (pp. 15-30). Edinburgh: Artificial Intelligence Applications Institute, University of Edinburgh.

[101] López, M. F., Gómez-Pérez, A., Sierra, J. P., & Sierra, A. P. (1999). Building a chemical ontology using methontology and the ontology design environment. IEEE Intelligent Systems and their applications, 14(1), 37-46.

[102] Grüninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies.

[103] Fernández-López, M., & Gómez-Pérez, A. (2002). Overview and analysis of methodologies for building ontologies. The Knowledge Engineering Review, 17(2), 129-156.

[104] Rojas, M.D. (1998). Ontologies Monatomic Ion in Physical Environmental Variables. Final-Year Project, Faculty of Informatics at the University of Madrid, (In Spanish).

[105] Gómez-Pérez, A., & Rojas-Amaya, M. D. (1999, May). Ontological reengineering for reuse. In International Conference on Knowledge Engineering and Knowledge Management (pp. 139-156). Springer, Berlin, Heidelberg.

[106] KBSI, 1994. The IDEF5 ontology description capture method overview. KBSI Report, Texas

[107] Jones, D., Bench-Capon, T., & Visser, P. (1998). Methodologies for ontology development.

[108] Farquhar, A., Fikes, R., Pratt, W., & Rice, J. (1995). Collaborative ontology construction for information integration. Technical Report KSL-95-63, Stanford University Knowledge Systems Laboratory, Retrieved from: http://citeseerx. ist. psu. edu/viewdoc/summary?doi=10.1.1.50.5303.

[109] Lenat, D. B., & Guha, R. V. (1989). Building large knowledge-based systems; representation and inference in the Cyc project. Addison-Wesley Longman Publishing Co., Inc..

[110] Swartout, B., Patil, R., Knight, K., & Russ, T. (1996, November). Toward distributed use of large-scale ontologies. In Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems (pp. 138-148).

[111] Mahesh, K., Helmreich, S., & Wilson, L. (1996). Ontology development for machine translation: Ideology and methodology. Computing Research Laboratory, New Mexico State University.

[112] Schreiber, G., Wielinga, B., & Jansweijer, W. (1995, August). The KACTUS view on the 'O'word. In IJCAI workshop on basic ontological issues in knowledge sharing (pp. 159-168).

[113] Sure, Y., Staab, S., & Studer, R. (2004). On-to-knowledge methodology (OTKM). In Handbook on ontologies (pp. 117-132). Springer Berlin Heidelberg.

[114] Gangemi, A., Steve, G., & Giacomelli, F. (1996, August). ONIONS: An ontological methodology for taxonomic knowledge integration. In Proceedings of the Workshop on Ontological Engineering, ECAI96.

[115] Bouaud, J., Bachimont, B., Charlet, J., & Zweigenbaum, P. (1994). Acquisition and structuring of an ontology within conceptual graphs. In Proceedings of ICCS (Vol. 94, pp. 1-25).

[116] Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.

[117] De Nicola, A., Missikoff, M., & Navigli, R. (2005). A proposal for a unified process for ontology building: UPON. In Database and Expert Systems Applications (pp. 655-664). Springer Berlin/Heidelberg.

[118] Sowa, J. F. (2000, August). Ontology, metadata, and semiotics. In ICCS (Vol. 1867, pp. 55-81).

[119] Yadav, U., Narula, G. S., Duhan, N., & Jain, V. (2016). Ontology Engineering and Development Aspects: A Survey. International Journal of Education and Management Engineering (IJEME), 6(3), 9.

[120] Lambrix, P. (2005, June). Towards a semantic web for bioinformatics using ontology-based annotation. In Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on (pp. 3-7). IEEE.

[121] Farooq, A., & Shah, A. (2008). Ontology Development Methodology for Semantic Web Systems. Pakistan Journal of Life Social Sciences, 6(1), 50-58.

[122] Mizoguchi, R. (2004). Tutorial on ontological engineering Part 2: Ontology development, tools and languages. New Generation Computing, 22(1), 61-96.

[123] Handschuh, S. (2007). Semantic web services: Concepts, technologies, and applications. Semantic Annotation of Resources in the Semantic Web, 135-155.

[124] Jain, V., & Singh, M. (2013). Ontology Development and Query Retrieval using Protutf-8. International Journal of Intelligent Systems and Applications, 5(9), 67.

[125] Preethi, M., & Akilandeswari, D. J. Combining Retrieval with Ontology Browsing. International Journal of Internet Computing (IJIC), 1.

[126] Abdolhamidzadeh, B., Abbasi, T., Rashtchian, D., & Abbasi, S. A. (2011). Domino effect in process-industry accidents–An inventory of past events and identification of some patterns. Journal of Loss Prevention in the Process Industries, 24(5), 575-593.

[127] Gauld, I. C., Giaquinto, J. M., Delashmitt, J. S., Hu, J., Ilas, G., Haverlock, T. J., & Romano, C. (2016). Re-evaluation of spent nuclear fuel assay data for the Three Mile Island unit 1 reactor and application to code validation. Annals of Nuclear Energy, 87, 267-281.

[128] Ahnlund, J., Bergquist, T., & Spaanenburg, L. (2003). Rule-based reduction of alarm signals in industrial control. Journal of Intelligent & Fuzzy Systems, 14(2), 73-84.

[129] Atzmueller, M., Arnu, D., & Schmidt, A. (2017). Anomaly Detection and Structural Analysis in Industrial Production Environments. In Data Science–Analytics and Applications (pp. 91-95). Springer Vieweg, Wiesbaden.

[130] Hernandez, J. Z., & Serrano, J. M. (2001). Knowledge-based models for emergency management systems. Expert Systems with Applications, 20(2), 173-186.

[131] Pham, S. B., & Hoffmann, A. (2003, December). A new approach for scientific citation classification using cue phrases. In Australasian Joint Conference on Artificial Intelligence (pp. 759-771). Springer, Berlin, Heidelberg.

[132] Han, S. C., Yoon, H. G., Kang, B. H., & Park, S. B. (2014). Using MCRDR based Agile approach for expert system development. Computing, 96(9), 897-908.

[133] Richards, D. (2009). Two decades of ripple down rules research. The Knowledge Engineering Review, 24(2), 159-184.

[134] Kang, B. H., Preston, P., & Compton, P. (1998, April). Simulated expert evaluation of multiple classification ripple down rules. In Proceedings of the 11th Workshop on Knowledge Acquisition, Modeling and Management.

[135] Bindoff, I., Kang, B. H., Ling, T., Tenni, P., & Peterson, G. (2007, December). Applying MCRDR to a multidisciplinary domain. In Australasian Joint Conference on Artificial Intelligence (pp. 519-528). Springer, Berlin, Heidelberg.

[136] Han, S. C., Mirowski, L., & Kang, B. H. (2015). Exploring a role for MCRDR in enhancing telehealth diagnostics. Multimedia Tools and Applications, 74(19), 8467-8481.

[137] Motameni, H., & Peykar, A. (2016). Morphology of compounds as standard words in persian through hidden Markov model and fuzzy method. Journal of Intelligent & Fuzzy Systems, 30(3), 1567-1580.

[138] Tjhai, G. C., Furnell, S. M., Papadaki, M., & Clarke, N. L. (2010). A preliminary two-stage alarm correlation and filtering system using SOM neural network and K-means algorithm. Computers & Security, 29(6), 712-723.

[139] Chen, M., Zheng, A. X., Lloyd, J., Jordan, M. I., & Brewer, E. (2004, May). Failure diagnosis using decision trees. In Autonomic Computing, 2004. Proceedings. International Conference on (pp. 36-43). IEEE.

[140] Gaines, B. R. (1989, December). An Ounce of Knowledge is Worth a Ton of Data: Quantitative studies of the Trade-Off between Expertise and Data Based On Statistically Well-Founded Empirical Induction. In ML (pp. 156-159).

[141] Gaines, B. R., & Compton, P. (1995). Induction of ripple-down rules applied to modeling large databases. Journal of Intelligent Information Systems, 5(3), 211-228.

[142] Cendrowska, J. (1987). PRISM: An algorithm for inducing modular rules. International Journal of Man-Machine Studies, 27(4), 349-370.

[143] Dietterich, T. (1995). Overfitting and undercomputing in machine learning. ACM computing surveys (CSUR), 27(3), 326-327.

[144] Joshi, M. V., & Kumar, V. (2004, April). Credos: Classification using ripple down structure (a case for rare classes). In Proceedings of the 2004 SIAM International Conference on Data Mining (pp. 321-332). Society for Industrial and Applied Mathematics.

[145] Kim, D., Han, S. C., Lin, Y., Kang, B. H., & Lee, S. (2018). RDR-based Knowledge Based

System to the Failure Detection in Industrial Cyber Physical Systems. Knowledge-Based Systems, 150, 1-13.

[146] Quinlan, J. R. (2014). C4. 5: programs for machine learning. Elsevier.

[147] Kim, D., Han, S. C., Lin, Y., Kang, B. H., & Lee, S. (2018). RDR-based Knowledge Based System to the Failure Detection in Industrial Cyber Physical Systems. Knowledge-Based Systems, 150, 1-13.

[148] Santos, I., Nieves, J., & Bringas, P. G. (2010, September). Enhancing fault prediction on automatic foundry processes. In World Automation Congress (WAC), 2010 (pp. 1-6). IEEE.

[149] Liu, Y., & Jiang, J. (2008, September). Fault diagnosis and prediction of hybrid system based on particle filter algorithm. In Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on (pp. 1491-1495). IEEE.

[150] Chen, B., Matthews, P. C., & Tavner, P. J. (2015). Automated on-line fault prognosis for wind turbine pitch systems using supervisory control and data acquisition. IET Renewable Power Generation, 9(5), 503-513.

Appendix A. List of Publications

Journal papers

International Journal Papers

[1] **Dohyeong Kim**, Soyeon Caren Han, Yingru Lin, Byeong Ho Kang and Sungsyoung Lee., (2018). RDR-based Knowledge Based System to the Failure Detection in Industrial Cyber Physical Systems. Knowledge-Based Systems (SCI, IF: 4.529), 150, pp.1-13.

[2] **Dohyeong Kim**, Yingru Lin, Byeong Ho Kang, Sungsyoung Lee, Soyeon Caren Han., (2018). A Hybrid Failure Diagnosis and Prediction using Natural Language-based Process Map and Rule-based Expert System. International Journal of Computers Communications & Control (SCIE, IF: 1.374), 13(2), pp.175-191.

[3] Taqdir Ali, Maqbool Hussain, Muhammad Afzal, Wajahat Ali Khan, Taeho Hur, **Dohyeong Kim**, M. Bilal Amin, Ho Jun Lim, Byeong Ho Kang and Sungyoung Lee. (2018). Clinically harmonized wellness concepts model for health and wellness services. IEEE Access (SCIE, IF: 3.244), 6, pp.26660-26674.

[4] Maqbool Ali, Rahman Ali, Wajahat Ali Khan, Soyeon Caren Han, Jaehun Bang, Taeho Hur, **Dohyeong Kim**, Sungyoung Lee, and Byeong Ho Kang. (2018). A Data-Driven Knowledge Acquisition System: An End-to-End Knowledge Engineering Process for Generating Production Rules. IEEE Access (SCIE, IF: 3.244), 6, pp.15587-15607.

[5] Thien Huynh-The, Cam-Hao Hua, Anh Tu Nguyen, Taeho Hur, Jaehun Bang, **Dohyeong Kim**, Muhammad B. Amin, Byeong Ho Kang, Hyonwoo Seung, Soo-Yong Shin, Eun-Soo Kim, Sungyoung Lee. (2018). Hierarchical topic modeling with pose-transition

feature for action recognition using 3D skeleton data. Information Sciences (SCI, IF: 4832), 444, pp.20-35.

[6] Thien Huynh-The, Cam-Hao Hua, Anh Tu Nguyen, Taeho Hur, Jaehun Bang, **Dohyeong Kim**, Muhammad Bilal Amin, Byeong Ho Kang, Hyonwoo Seung and Sungyoung Lee. (2018). Selective bit embedding scheme for robust blind color image watermarking. Information Sciences (SCI, IF: 4832), 426, pp.1-18.

[7] Muhammad Asif Razzaq, Claudia Villalonga, Sungyoung Lee, Usman Akhtar, Maqbool Ali, Eun-Soo Kim, Asad Masood Khattak, Hyonwoo Seung, Taeho Hur, Jaehun Bang, **Dohyeong Kim** and Wajahat Ali Khan. (2017). mlCAF: Multi-Level Cross-Domain Semantic Context Fusioning for Behavior Identificatio-n. Sensors (SCIE, IF: 2.677), 17(10), 2433.

[8] Taeho Hur, Jaehun Bang, **Dohyeong Kim**, Oresti Banos and Sungyoung Lee. (2017). Smartphone Location-Independent Physical Activity Recognition Based on Transportation Natural Vibration Analysis. Sensors (SCIE, IF: 2.033), 17(4), 931.

Domestic Journal Papers (Korean, KCI)

[9] **Dohyeong Kim**, Byeong Ho Kang and Sungyoung Lee. (2018). Failure Knowledge Extraction Framework from Failure Reports in Large Industries. Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, 8(3). pp.955-964.

[10] **Dohyeong Kim**, Byeong Ho Kang and Sungyoung Lee. (2017). Hot Search Keyword Rank-Change Prediction. Journal of KIISE, 44(8), pp.782-790.

[11] **Dohyeong Kim**, Byeong Ho Kang and Sungyoung Lee. (2017). Preventive Maintenance System based on Expert Knowledge in Large Scale Industry. KIISE Transactions on Computing practices, 23(1). pp.1-12.

Conference papers

International Conference Papers

[12] **Dohyeong Kim**, Soyeon Caren Han, Sungyoung Lee and Byeong Ho Kang. (2016, December). Predicting the Rank of Trending Topics. In Australasian Joint Conference on Artificial Intelligence (pp. 636-647). Springer, Cham.

[13] **Dohyeong Kim**, Soyeon Caren Han, Sungyoung Lee and Byeong Ho Kang. (2016, August). Predicting the scale of trending topic diffusion among online communities. In Pacific Rim Knowledge Acquisition Workshop (pp.153-165). Springer, Cham.

Domestic Conference Papers (Korean)

[14] **Dohyeong Kim,** Byeong Ho Kang and Sungyoung Lee. (2017). Failure Knowledge Extraction Framework from Failure Reports, HSST, pp.1-4

[15] **Dohyeong Kim** and Sungyoung Lee. (2017). Failure prediction system based on expert knowledge, KIPS, pp.945-947.

[16] **Dohyeong Kim** and Sungyoung Lee. (2016). Information Extraction Framework from Failure Report, KIISE. pp.545-587.

[17] Sang-Ho Lee, **Dohyeong Kim**, Sungyoung Lee, Byeong Ho Kang. (2014), Web contents mashup service framework with Crawler web page and Classification engine, KIISE, pp.1872-1874.

[18] **Dohyeong Kim**, Sungyoung Lee, Tae-choong Jung. (2013). Study on A Personal Service Recommendation Method Based on MCRDR under Smartphone, KIISE, pp.482-483.

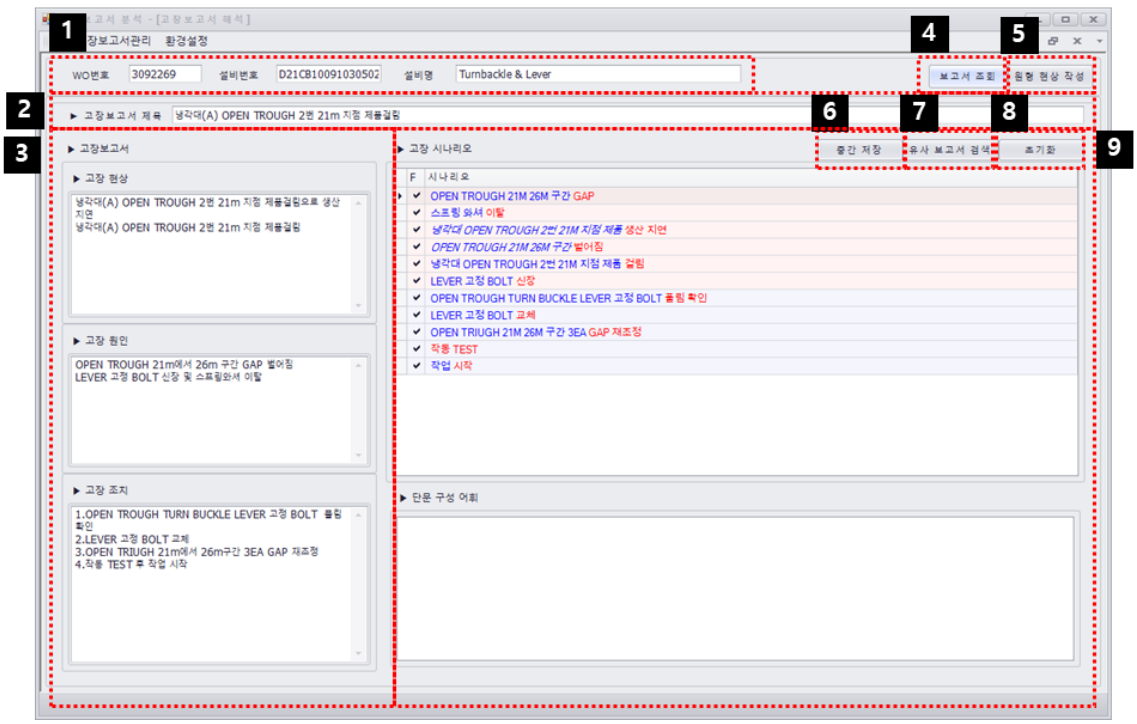
Appendix B. Introduction for Failure Report Analysis System

B.1 Main view

B.1.1 Introduction

This is the failure report interpretation screen.

This screen shows the contents of the original failure report and the failure scenarios analyzed on the basis of it, and it provide a function which enables to modify the failure scenarios automatically analyzed by the system through user feedback.



B.1.2 Summary

If the user selects the ‘Failure Report Analysis’ menu, it provides that the user can see the contents and the scenario of the failure report on the displayed window for analyzing the failure report.

B.1.3 Functions

① Displays the Work Order Number (WONUM as failure report ID) of the failure report and the target failure ID and facility name.

② Displays the title of the failure report to be interpreted.

③ Displays the contents of the failure report to be analyzed.

④ Provides a function to search the list of failure reports to be analyzed.

⑤ Provides the ability to move to the next step for the interpretation of failure reports.

⑥ Provides the function to save the current contents.

⑦ Provides the ability to search for previously analyzed failure reports that are similar to the failure reports currently being interpreted. The contents of the selected similar reports should be reused for analysis of the current report.

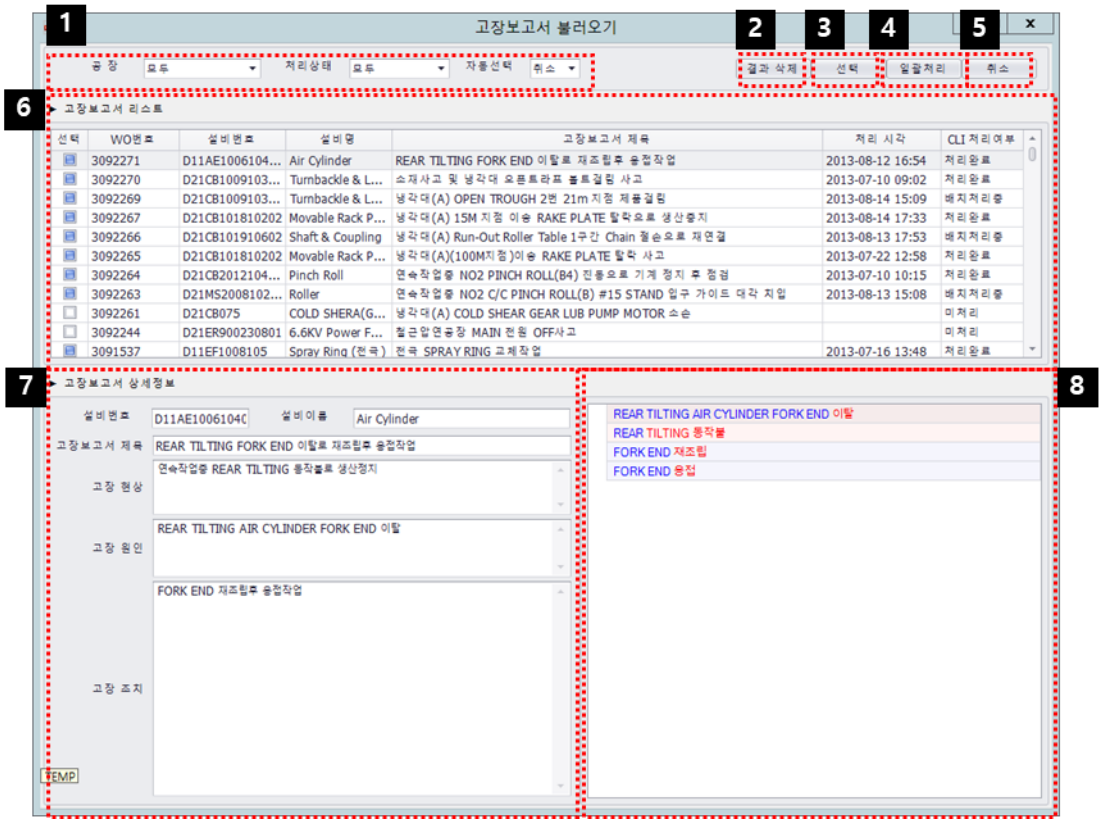
⑧ Discards all content that has been processed so far current and provides to return into the initial contents that are automatically interpreted by the system.

⑨ Provides functions for displaying and correcting the failure scenario automatically interpreted by the system.

B.2 Failure report loading window

B.2.1 Introduction

This is the failure report interpretation screen.



B.2.2 Summary

If the user selects the ‘Failure Report Analysis’ menu, it provides that the user can see the contents and the scenario of the failure report on the displayed window for analyzing the failure report.

B.2.3 Functions

① The function of filtering the failure report list according to specific factory, processing condition and automatic monitoring for batch processing. Provides the ability to select a book.

② Provide the function of filtering the failure report list according to specific factory, processing status and selecting automatically the failure report for batch processing.

③ If the report selected in ⑥ has already been interpreted, it provides the function to delete interpreted and stored in the process map.

④ Provide the function to confirm for analyzing the selected failure report

⑤ Provide the function to batch processing multiple failure reports selected by check-box

⑥ Exit the failure report loading window.

⑦ Provides a function to output a list of Failure reports and select a Failure report to interpret. On the left, it provides a function to select multiple Failure reports by check box and batch process.

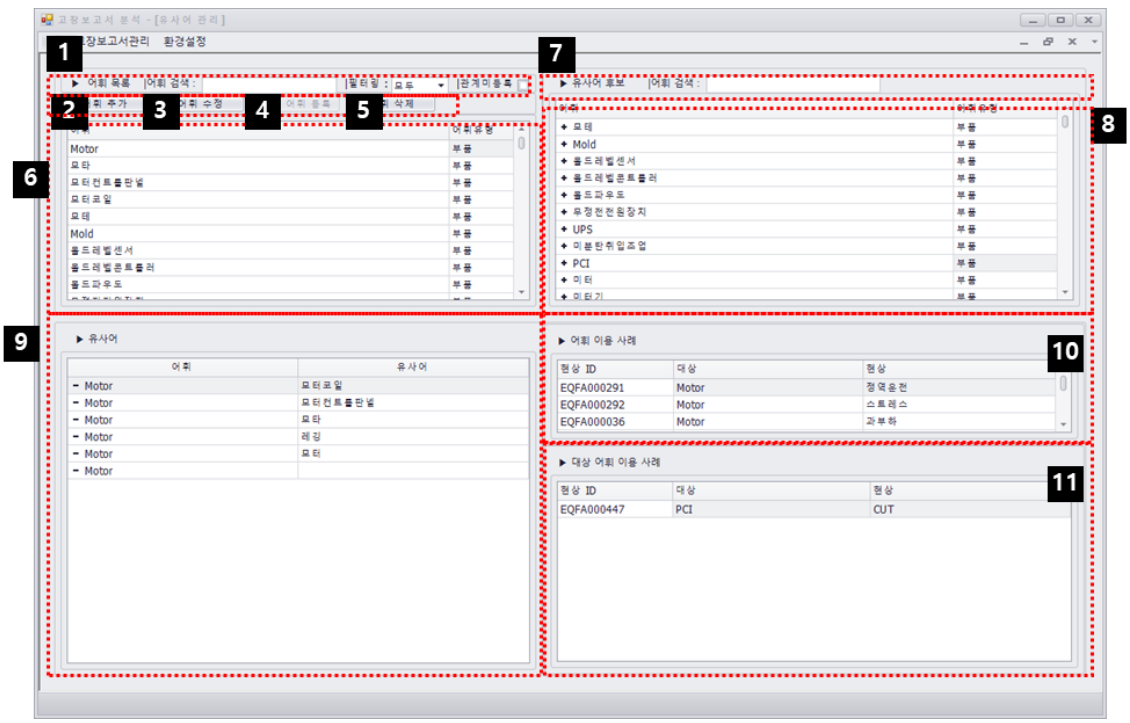
⑧ Print the contents of the selected Failure report.

⑨ If the selected Failure report is already analyzed, the fault scenario is output.

B.3 Similar vocabulary management

B.3.1 Introduction

This is a screen for managing the vocabulary stored in the system and the similar relation between them. It displays a list of all the vocabulary stored in the system, a vocabulary candidate list for similarity to each vocabulary, and a list of similar vocabulary associations that are already connected. In addition, each vocabulary displays the contents of the used simple sentences to help understand the meaning of each vocabulary.



B.3.2 Summary

This picture shows followings.

- A list of the vocabulary to establish a similarity relation
- A candidate vocabulary which a relationship can be established,
- A list of vocabulary that already has a similar relationship with each vocabulary
- A simple sentence which two vocabularies selected for the similarity setting

B.3.3 Functions

① Provides vocabulary search and word type-based filtering

② Provide the ability to add new vocabulary

③ Provide the ability to modify the selected vocabulary

④ If the selected vocabulary is a user analysis vocabulary,
it provides the ability to register as a standard vocabulary

⑤ Provide the function to delete vocabulary

⑥ Display a list of the words stored in the system

⑦ Vocabulary that can be established similar to the selected vocabulary in 6 is displayed

⑧ If the '+' button is clicked, it is registered as a similar word

⑨ A list of similarity vocabularies of the selected vocabulary is displayed. If the '-' button is clicked, the analogy relationship is released

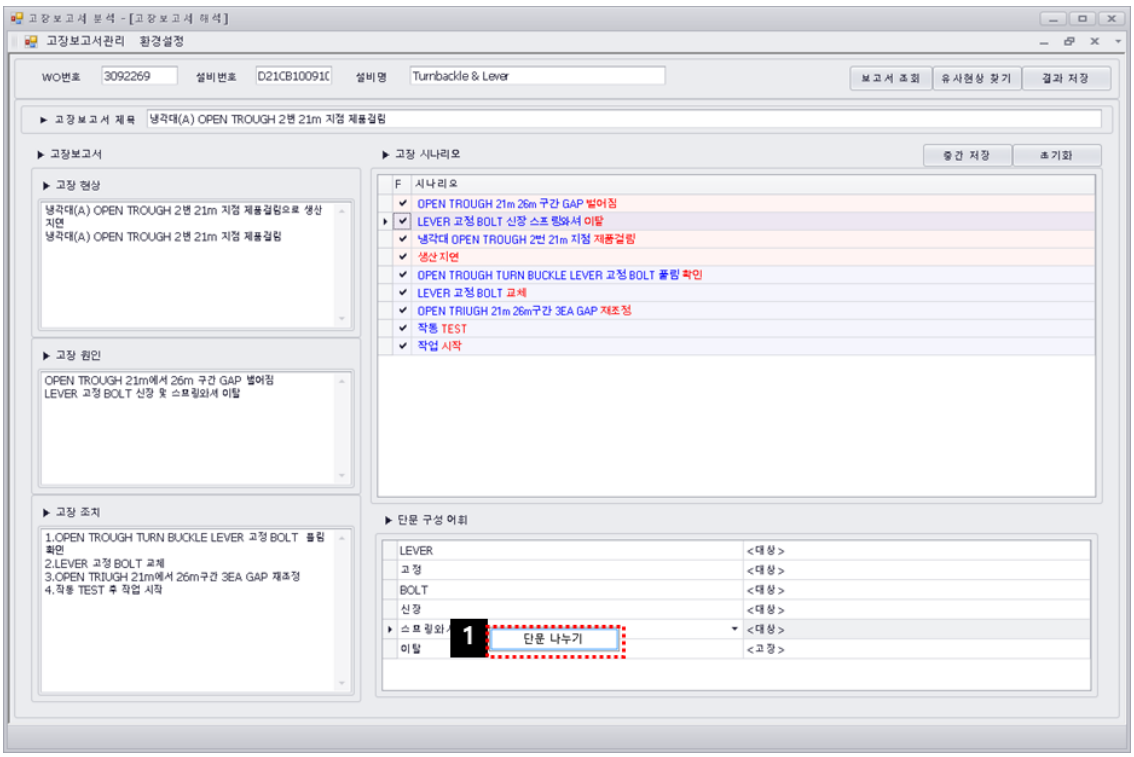
⑩ A list of short sentences in which the selected vocabulary is selected in step # 6 is displayed

⑪ A list of short sentences in which the selected vocabulary is selected in step # 8 is displayed

B.4 Sentence Separation

B.4.1 Introduction

It is a screen to modify the contents of a short sentence that constitutes a failure scenario.



B.4.2 Summary

It provides a function to divide one short sentence into two short sentences.

B.4.3 Functions

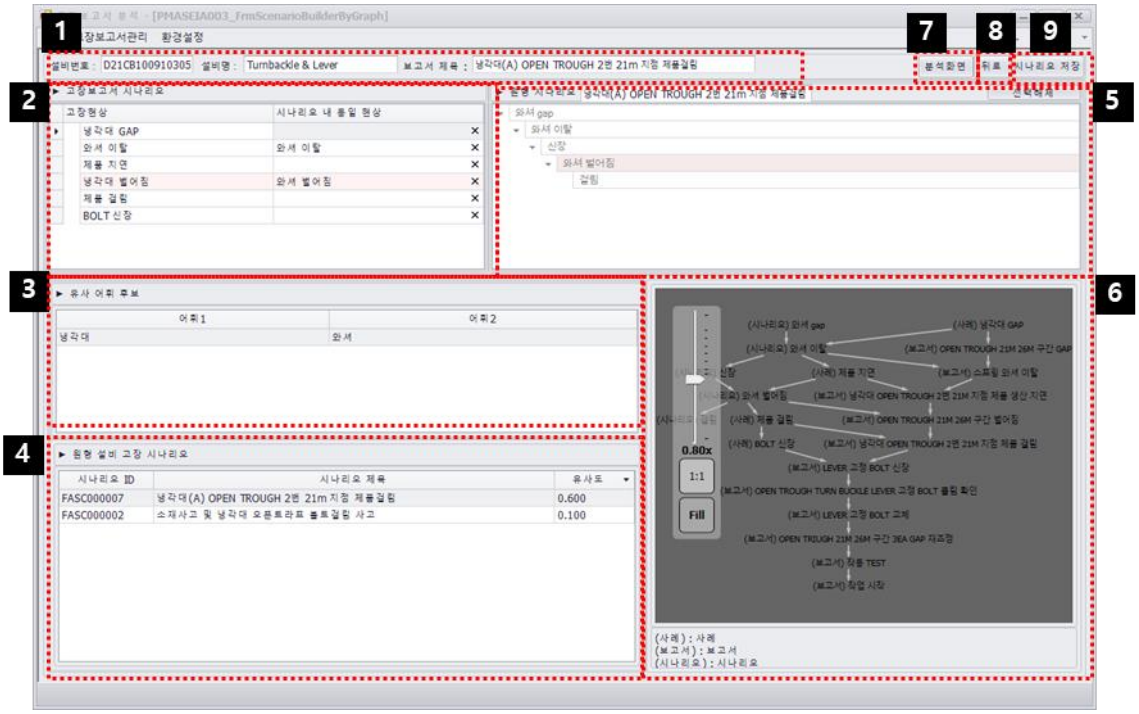
① Separate the selected short sentences by the selected vocabulary into two short sentences

consisting of the short vocabulary consisting of the preceding vocabulary and the selected vocabulary and the following vocabulary.

B.5 Failure Scenario Creation

B.5.1 Introduction

This view creates a failure case scenario based on the contents of the analyzed failure report. And, build case scenarios that consist of only the currently analyzed reports or build scenarios that are integrated with existing and similar scenarios.



B.5.2 Summary

The content of the case scenario based on the failure report currently being analyzed is displayed. In addition, a list of pre-prepared case scenarios is output from the same facility, and the contents of these scenarios are output. In addition, a list of similar vocabularies newly obtained from the similarity relationship between the existing scenarios and the case scenarios currently being reanalyzed and the scenario contents to be finally stored are drawn in a graph structure.

B.5.3 Functions

① Output the target facility and title of the failure report currently being analyzed.

② Output the case scenario generated from the failure report currently under analysis and the similar phenomenon of each phenomenon (from the existing case scenario).

③ A list of similar vocabulary newly obtained from similar phenomena appears.

④ A list of similar case scenarios in the report currently under analysis appears.

⑤ The contents of the case scenario selected in screen 4 are displayed. By dragging each node to the phenomena in screen 2, a similar phenomenon relationship can be generated.

⑥ The structure of the case scenario to be stored finally is displayed in graph form.

⑦ Move to the failure report analyze as the first screen.

⑧ Go to scenario creation screen.

⑨ Save the contents of the analyzed failure report and case scenario.

Appendix C. Case Study: Alarm Data

Raw alarm data sample:

	① CRE_DT	② CRE _TIME	③ FCTY _CD	④ EQP _NUM	⑤ SNSR_TAG_ID
1	20131125	23	H4A2	H1102396	ALM_PRC_FE_020
2	20131125	23	H4A2	H1102396	ALM_PRC_FE_007
3	20131125	23	H4A2	H1102396	ALM_PRC_FE_008
4	20131125	23	H4A2	H1102396	ALM_PRC_FE_018
5	20131125	22	H1C1	H1072672	C112BC_HALMSTOP_MD
6	20131125	22	H1C1	H1072672	C112BC_HALMSTOP_PULLCORD_GR01
7	20131125	22	H1C1	H1072672	C123BC_HALMSTOP_PULLCORD_GR02
8	20131125	22	H1C1	H1072756	K153BC_HALMSTOP_PULLCORD_GR02
9	20131125	22	H2B1	H1055078	FROL.R28B_PLR021
10	20131125	22	H2B1	H1053976	LDSG.R18B_LDSG054
11	20131125	22	H2B1	H1053976	LDWG.R1B_ARM24_PHA
12	20131125	22	H2B1	H1053976	LDWG.R1B_ARM14_PHHA
13	20131125	22	H2B1	H1053976	LDWG.R1B_ARM13_PHA
14	20131125	22	H2B2	H1055927	FROL.R28B_DBR117
15	20131125	22	H1A1	H1000144	B103_LOADBEARING_3

	⑥ DIV	⑦ DIV_LEN	⑧ ALRM_GRD	⑨ UNUSL_ALRM	⑩ CRE_ZONE
1	H	24	10	Y	*NULL*
2	H	24	10	Y	*NULL*
3	H	1	10	Y	4
4	H	1	10	Y	1
5	H	1	10	Y	1
6	H	24	10	Y	4
7	H	24	1	Y	4
8	H	1	1	Y	4
9	H	1	1	Y	4
10	H	1	1	Y	*NULL*
11	H	1	10	Y	4
12	H	24	10	Y	*NULL*
13	H	1	10	Y	*NULL*
14	H	24	10	Y	*NULL*
15	H	24	10	Y	*NULL*

	⑪ CRE_CNT	⑫ CNTN_TIME	⑬ AVG_ALRM_GAP	⑭ AVG_CNTN_TIME	⑮ ALRM_CRE_RATE	⑯ ALRM_IDX	⑰ ALRM_RATE
1	284	2369	295.8838028	8.96969697	8.222222222	862.953995	0.009987893
2	38	209	2268.184211	5.8	0.805555556	120.974517	0.001400168
3	34	190	2535.588235	5.75	0.638888889	96.6172771	0.001118256
4	5	25	17275	5.5	0.305555556	48.1471162	0.000557258
5	7	2509	11984.42857	444	12.33333333	27.3764259	0.000316857
6	4	721	21419.75	180.25	3.527777778	24.8776274	0.000287936
7	1	140	86260	140	3.888888889	24.9710983	0.000289017
8	3	584	28605.33333	194.6666667	5.083333333	25.285338	0.000292654
9	4	534	21466.5	134	3.722222222	24.9278707	0.000288517
10	38	4154	2164.368421	109.3157895	31.58333333	526.187576	0.006090134
11	19	1247	4481.736842	65.63157895	13.97222222	223.183726	0.002583145
12	11	426	7815.818182	38.72727273	4.055555556	100.057904	0.001158078
13	19	707	4510.157895	63	10.5	160.893855	0.001862197
14	23	42045	1928.478261	1828.043478	51.94444444	99.8843931	0.001156069
15	4	8772	19407	3600	100	86400	1

Total attributes of raw data: 17
Selected attributes for training dataset: 6

Num.	Attribute	Description
①	CRE_DT	Occurrence date of the alarm
②	CRE_TIME	Occurrence time of the alarm
③	FCTY_CD	Code of factory occurring alarm
④	EQP_NUM	Code of facility occurring alarm
⑤	SNSR_TAG_ID	ID of the alarm
⑥	DIV	Alarm class
⑦	DIV_LEN	Alarm class duration
⑧	ALARM_GRD	Alarm grade
⑨	UNUSL_ALRM	Unusual occurring alarm
⑩	CRE_ZONE	Alarm Creation Zone
⑪	CRE_CNT	the occurrence of the alarm data in one hour
⑫	CNTN_TIME	the length of time that the specific alarm is alive in one hour
⑬	AVG_ALRM_GAP	Average creation gap of Alarm
⑭	AVG_CNTN_TIME	Average Counts of Alarm
⑮	ALRM_CRE_RATE	Creation rate of alarm
⑯	ALRM_IDX	Alarm Index
⑰	ALRM_RATE	the percentage of resources taken by the specific alarm